Scale Mixtures Variational Autoencoder

Kontrasztinvariáns reprezentáció neurális hálóban

— SZAKDOLGOZAT —

Készítette:

Martos Domonkos

Matematika BSc

Alkalmazott matematikus szakirány

Témavezetők:

Orbán Gergő

Wigner Fizikai Kutatóközpont

Lendület Kutatócsoport

Prokaj Vilmos

Valószínűségelméleti és

Statisztika Tanszék

Eötvös Loránd Tudományegyetem

Természettudományi Kar

Budapest, 2023

Tartalomjegyzék

1. Bevezetés

A *deep learning*, azaz mélytanulási modellek egyre nagyobb teret nyertek az elmúlt években. A képfeldolgozás terén a konvolúciós neurális háló (CNN) jelentett áttörést, ami diszkriminatív feladatokat old meg hatékonyan (azaz a bemeneti képhez a megfelelő címkét igyekszik kiválasztani). A modell a képeken lévő különböző mintázatokat és textúrákat tanulja, így többféle invariancia, például eltolás vagy tükrözés mellett is képes meghatározni ugyanazt a kategóriát. Ezek a sikerek felügyelt tanulás (*supervised learning*) mellett születtek, vagyis a modellnek szüksége van egy helyes címkékkel ellátott adathalmazra a tanuláshoz. Ez egyrészt költségessé teszi a megfelelő tanító adathalmazok létrehozását, másrészt nem egyezik meg egy valós tanulási szituációval. A biológiai rendszerek címkék nélkül alakítanak ki egy reprezentációt a kapott információk alapján, ez pedig arra inspirál, hogy az önálló tanulás (*unsupervised learning*) keretrendszerében dolgozó modellt válasszunk.

Ez a célkitűzés irányít a mélygeneratív modellek irányába. Ezekben a tanult reprezentációt egy látens tér dimenziói jelenítik meg, ezzel pedig könnyebben alakítható és vizsgálható a CNN-nél. Az önálló tanulás során a modell felfedezi az adathalmazbeli képek mögöttes eloszlását, és így a folyamat során egy generatív komponenst is kapunk. A célom annak vizsgálata, hogy a látens térben megjelenő generatív elemek milyen hierarchia szerint rendeződnek el. Illusztrációként gondoljuk el, hogy egy kupac száraz bükkfalevél mintázatát szeretnénk generálni. Azt szeretnénk, hogy a generatív elemek egy része az egyes levelek mintázatát tanulja meg, és más részük legyen felelős a levelek elhelyezkedéséért. A komponenseket azonban nehéz matematikailag szétválasztani, hiszen egy kellően flexibilis, nemlineáris modell hatékonyan képes tanulni enélkül is - így viszont elveszik a mintázat struktúrájának mélyebb megértése.

Azt, hogy ilyen hierarchikus látens teret alkosson a modell, a felépítésében jelenlévő induktív torzítással (*inductive bias*) tudjuk segíteni. Ezzel be tudjuk építeni a modellbe a tanulandó adathalmazról való előzetes tudásunkat, ezzel segítve a tanulást. Egészítsük ki a faleveles példát egy egyszerű fizikai invarianciával, a kontraszttal. A levelek egy adott elrendezése különböző fényviszonyok mellett ugyanaz, azonban kisebb kontraszt nagyobb bizonytalanságot, azaz varianciát jelent az elrendezésben. Egy ilyen kép esetében a kontraszt egy fényerősséget jelent, amit egy adott elrendezés esetében a kép pixeleire vett szorzás ír le. Ebben az esetben tehát a modellben ezt a szorzást megragadó induktív torzítást kell bevezessünk, hogy segítsük a struktúra megértését.

Az induktív torzítás jól megvalósítható a Variational Autoencoder (VAE) nevű mélygeneratív modell esetében. Ez a modell azért is vonzó, mert a bayesi statisztika elvei szerint működik, ezáltal pedig átlátható garanciák és megkötések vonatkoznak rá. Ezáltal egy normatív keretben vagyunk képesek következtetéseket levonni,

ami más generatív modellek esetében nem adott. A dolgozatom célja, hogy egy kontrasztinvariáns látens reprezentációval rendelkező modellt alkosson a VAE keretrendszerében. A tanítás a MNIST adathalmaz kontraszttal augmentált változatán történik. A távolabbi cél az, hogy természetes képeken működjön a modell, azonban az ezekben lévő összetettebb struktúra miatt az erre való modellépítés idő- és erőforrásigényes. Emiatt fontos, hogy ezelőtt felfedezzük, hogy milyen modell képes hatékony tanulásra. A dolgozatban három különböző felépítésű modellt vizsgálok. Az egyikük egy standard VAE, amiről azt találom, hogy jól tanulja a kontrasztot, azonban nem egy külön látens dimenzióban jeleníti meg azt. Ezért két különböző továbbfejlesztést valósítok meg, amiben egy látens koordináták közötti szorzás hivatott szétválasztani jelentés szerint a látens dimenziókat - ez a *pre hoc* és *post hoc* Scale Mixtures Variational Autoencoder. A vizsgálat során azt találom, hogy a *post hoc* SMVAE képes leginkább kontrasztinvariáns reprezentáció és helyes rekonstrukció tanulására.

A bevezetés után a második fejezetben röviden bemutatom a CNN felépítését, majd a GAN nevű generatív modellt. A fejezet végén részletesen tárgyalom a VAE felépítését. Ezután a harmadik fejezetben leírom a Gaussian Scale Mixtures nevű, valószínűségi eloszlásokat vegyítő ötletet és motivációját, ami a megvalósított modellekben bevezetett szorzást inspirálta. A negyedik fejezetben a felhasznált adathalmaz bemutatása után részletezem a modellek felépítését és összefoglalom az eredményeket. A Standard VAE, illetve a *pre hoc* és *post hoc* SMVAE esetében is bemutatom a látens teret, a rekonstrukciót, illetve vizsgálom a kontraszt elkódolásának függetlenségét. Végül az ötödik fejezetben összefoglalom az eredményeket, majd kitérek a limitációkra és továbbfejlesztési lehetőségekre.

2. Mélytanulási modellek a képfeldolgozásban

2.1. Neurális háló képklasszifikációra

A mélytanulási modellek jelentős részének alapja az a mesterséges neurális hálózat struktúra, amit már régen kidolgoztak, azonban a számítási kapacitások és felhasználható adathalmazok az előző évtizedre nőttek meg annyira, hogy a gyakorlatban is jól alkalmazható legyen. A mesterséges neurális hálókról és alkalmazásaikról ír Bishop (1994) harminc évvel ezelőtt összefoglalót. A *mesterséges* és *neurális* szavak itt arra utalnak, hogy ezek a modellek a biológiai neurális hálózatokról vesznek mintát.

Egy mesterséges neurális háló tulajdonképpen egy $f : \mathbb{R}^n \to \mathbb{R}^m$ függvény, ami egy speciális struktúra szerint épül fel. A bemenet $x \in \mathbb{R}^n$ vektort elsőként egy k dimenziós, úgynevezett rejtett rétegbe transzformálja a háló. Ennek egy a_j koor-

dinátájának (egy "mesterséges neuron") értékét az előző réteg, azaz jelen esetben a bemenet koordinátáinak súlyozott összege alapján fejezzük ki. Vagyis egy a_j koordináta értékét a $\sum_{i=1}^{n} w_{ji}x_i$ kifejezés szerint kapjuk meg, ahol $w_j \in \mathbb{R}^n$ az a_j koordinátára nézve egyedi súlyozás. Hogy ne csak lineáris összefüggések legyenek a modellben, ez az érték még egy nemlineáris, úgynevezett *aktivációs függvény*en is átmegy, azaz $\forall j \in 1 \dots k : \ a_j = g\left(\sum_{i=1}^{n} w_{ji}x_i\right)$, ahol g a nemlineáris aktivációs függvény (azaz a "neuron" akkor "tüzel", ha elég nagy bemeneti impulzust kap). Egy egész réteg neuronértékeinek kiszámítását mátrixszorzással végezhetjük:

$$a = g(Wx)$$

ahol $a \in \mathbb{R}^k$ egy rejtett réteg k neuronjának aktivációi, W mátrix tartalmazza az adott rejtett réteg súlyait (a j. sor i. eleme, w_{ji} az a_j koordináta képletében x_i súlya), x pedig az előző réteg. Rejtett rétegekből követi egymást több, majd végül megkapjuk a modell $f(x) \in \mathbb{R}^m$ kimenetét. Az aktivációs függvényekre néhány példa:

$$\begin{aligned}
\text{ReLU: } g(x) &= \max(0, x) \\
\text{sigmoid: } g(x) &= \frac{1}{1 + e^{-x}} \\
\text{softplus: } g(x) &= \log(1 + e^{x})
\end{aligned}$$

A cél az, hogy a w paramétereket optimálisan állítsuk be, azaz a feladatunknak megfelelően működjön a háló. A jó paraméterek megtalálásához egy optimalizálási algoritmusra van szükség - ez a *backpropagation*, azaz a hiba-visszaterjesztési algoritmus szerint valósul meg. Ehhez egy megfelelő veszteségfüggvényt kell definiálni, majd az algoritmus eszerint kiértékeli $f(x)$-et, és változtat a paramétereken. Erről részletesebben a (2.3.2) részben írok. Az eredményes tanuláshoz legtöbbször sok rejtett rétegre, és megfelelően nagy adathalmazra, amin az optimális paramétereket meg tudja tanulni a háló.

Az első nagy áttörést Krizhevsky, Sutskever, és Hinton (2012) munkája hozta, akik képklasszifikációs feladatra építettek fel egy 60 millió paraméterrel rendelkező speciális hálót. Egy konvolúciós neurális hálót (*convolutional neural network*, CNN) hoztak létre, ami az ImageNet nevű, természetes képeket és címkéiket tartalmazó adathalmaz elemeit "ismeri fel", azaz minden képhez egy valószínűségi eloszlást rendel a címkékre nézve. Az ehhez hasonló, *diszkriminatív* modellek esetében a kimenet előtt található egy normalizáló réteg, ami az utolsó előtti réteg értékeit valószínűségi eloszlássá transzformálja (azaz kimenete minden koordinátában $(0, 1)$-beli, és a koordináták összege 1). Ez jellemzően a *softmax* függvény:

$$\sigma : \mathbb{R}^k \to (0,1)^k, \ \sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \quad (i \in 1 \dots k)$$

Krizhevsky és társainak hálózata az esetek 83%-ában az 5 legvalószínűbb címke közé sorolta a helyes megoldást az 1000 lehetséges címke közül. Ez az arány akkor figyelemreméltóan magasabb volt más eljárások pontosságánál. Munkájuk már korábban is létező eljárásokat valósított meg egy új kombinációban, és megmutatta, hogy a mélytanulásban valós lehetőség rejlik. Eredményeik nyomán rengetegen dolgoztak különböző neurális hálózatok fejlesztésén (az eredeti cikk jelenleg 133 ezer hivatkozás körül jár a Google Scholar szerint).

A CNN architektúra több változtatást eszközöl a fentebb leírt, alap neurális háló struktúráján - a képek esetében ugyanis kulcsfontosságú a különböző textúrák és tulajdonságok megtanulása, amik a vektorrá alakított reprezentációban elvesznek. A CNN ezért kétdimenziós konvolúciós rétegeket használ, amik ugyanúgy az előző réteg néhány elemének lineáris kombinációja szerint adják az új réteg elemeit. Itt azonban a súlyok előre meghatározottak és fixek, valamint az új réteg egy elemére az előző réteg csak néhány, a 2D térben közeli pixel értéke fog hatni. Egy konvolúciós réteg így egy "ablakot" csúsztat végig a kétdimenziós inputon. A modell a konvolúciós rétegek mellett dimenziócsökkentő (*pooling*) rétegeket használ, és ezekkel hatékonyan tud magasabb és alacsonyabb szintű mintázatokat, textúrákat tanulni. A megfelelő reprezentáció kialakítása után kell vektorizálni, "kilapítani" az adatokat, és itt néhány hagyományos, illetve egy softmax réteg oldja meg a klasszifikáció feladatát.

A CNN tehát egy hatékony megoldás diszkriminatív feladat esetén, felügyelt tanulás (*supervised learning*) mellett. A modell invariáns a transzlációra, vagyis egy kép eltoltját, vagy tükrözöttjét is felismeri, hiába teljesen mások a konkrét pixelértékek a két képen. Ez a felépítés tehát egy olyan induktív torzítást jelent, ami a segíti a transzlációs invariancia tanulását. Azonban a tanuláshoz szükség van arra, hogy a tanító adathalmaz tartalmazza a helyes címkéket minden kép esetén, a modell veszteségfüggvénye pedig ezt, a helyes választ veti össze a kimenetként kapott valószínűségi eloszlással. Ennek egyértelmű hátránya, hogy költségesebb egy ilyen adathalmaz összeállítása. Az eredeti hálónak sok különböző irányú továbbfejlesztése született az azóta eltelt években, azonban az önálló tanulás és a képek eloszlásának tanulására nem kifejezetten alkalmas ez a modell (Gu és tsai., 2018). Másrészt, mivel a modell x bemenetre egy $p(z|x)$ valószínűségi eloszlást számol (ahol $z \in \mathbb{R}^m$ a kimeneti vektor), a képek $p(x)$ eloszlásának tulajdonságairól nem tudunk tanulni általa. A közbülső rétegekben észrevehetünk mintázatokat aszerint, hogy milyen textúrát hol kódol el a hálózat, azonban végső soron a modell címkék között dönt. Ez pedig nem egyezik meg egy biológiai hálózat működésével, ahol fontosabb a látottak értelmezése a címkézésnél. Továbbá a strukturális értelmezés és flexibilis inferencia is hiányzik: például egy állat esetében nem csak az érdekes, hogy milyen állat van a képen, hanem a viselkedése, mint hogy milyen irányba mozoghat, veszélyes-e.

2.2. Generatív modellezés: a GAN

Az önálló tanulás (*unsupervised learning*) esetében a modellnek csak a feldolgozandó adathalmazra van szüksége, címkék és egyéb kiegészítések nélkül. Feltételezzük, hogy az adathalmaz x elemei egy ismeretlen mögöttes eloszlásból származnak, legyen ez $p^*(x)$. A célunk egy olyan modell megalkotása, ami ezt az eloszlást közelíti, azaz kell:

$$p_\theta(x) \sim p^*(x)$$

A közelítést *maximum likelihood* becsléssel javíthatjuk, azaz a lehetséges paramétereket tartalmazó Θ halmazban keressük azt a paraméterezést, amelyik a legjobban közelíti az ismeretlen eloszlást, azaz keressük:

$$\{p_\theta\}_{\theta\in\Theta} : \min_{\theta\in\Theta} L(p_\theta, p^*) \tag{2.1}$$

egy adott $L : \mathbb{R}^n \to \mathbb{R}$ veszteségfüggvényre. Ha p_θ jól közelíti p^*-ot, akkor p_θ-ból mintát véve a tanító adathalmaz pontjaihoz hasonló adatpontokat generálhatunk. A mögöttes $p^*(x)$ eloszlást azonban nem biztos, hogy egy explicit paraméterű eloszlással le lehet írni, ezért implicit módon szeretnénk közelíteni az adathalmaz struktúráját.

Ennek egy népszerű módszere Goodfellow és tsai. (2014) munkája nyomán a *Generative Adversarial Network* (GAN). A módszer sokféleképpen használható, most képekre vonatkoztatva írok róla. Az architektúra felépítése játékelméleti alapon nyugszik: két gépi tanulási modell, a Generátor és a Diszkriminátor (amik jellemzően neurális hálók) "versenyeznek" egymással az optimalizáció során. A Generátor feladata, hogy az eredetihez hasonló képeket generáljon. Ehhez egy sokdimenziós z inputot kap, ami egy adott prior szerinti zaj (például $p(z) = \mathcal{N}(0, I)$). Ebből állítja elő a $G(z; \theta_G)$ kimenetet (ahol θ_G a paramétereket jelenti). A Diszkriminátor feladata, hogy egy valódi x kép és $G(z; \theta_G)$ közül kiválassza az igazi képet. A tanuláshoz mindkét rész kap egy költségfüggvényt: $L_G(\theta_G, \theta_D)$ és $L_D(\theta_G, \theta_D)$. A Diszkriminátor esetében ez lehet egy egyszerű, bináris klasszifikációra szolgáló negatív log likelihood függvény: $\log(y)$, ahol y a helyes címkére tippelt valószínűség. A Generátoré pedig vagy $-L_D$ (ekkor egy minimax algoritmust kapunk), vagy pedig $\log(1 - y)$ (azaz L_D fordított címkékkel: az a cél, hogy rossz kategóriába sorolja a Diszkriminátor). A paraméterek frissítése gradiens módszerrel történhet.

A tanulási folyamat elméletben addig tart, amíg egy lokális Nash-egyensúly áll fenn a két rész között. Ez azt jelenti, hogy egyik sem tudja úgy frissíteni a paramétereit egy adott részhalmazon, hogy jobb eredményt érjen el a másik paramétereinek változatlansága mellett. Ekkorra a Generátor implicit megtanulja, közelíti az eredeti képek $p^*(x)$ eloszlását. A modell egy fix látens térrel rendelkezik z felett, vagyis különböző z vektorokat adva bemenetként a Generátornak, felfedezhetjük, hogy milyen reprezentációt tanult meg. Izgalmas tulajdonsága ennek a térnek, hogy egy generált

képen lévő szemantikai változásnak gyakran egy adott irányú látens vektor felel meg - így például egy emberi arc generálására tanított GAN látens terében megtalálhatjuk a mosolynak megfelelő irányt, és egy adott arc z ősképét ebbe az irányba tolva mosolyt csalhatunk rá. Ilyen irányok tartalmazhatnak forgatást, fényerőre, háttérre vonatkozó információkat is. A modell tehát implicit felfedez szemantikus tulajdonságokat a látens reprezentációban. Ezek szétválasztása (*disentanglement*) azonban nem a látens koordináták mentén valósul meg, hanem összetett irányokban jelenik meg, és a modell struktúrája miatt ezek tanulását nem tudjuk induktív torzítással segíteni.

A GAN jó eredményeket képes elérni a képgenerálás terén, azonban több hátránya is van. Az egyik a tanítás nehézkessége. A lokális Nash-egyensúly megtalálása nem sikerül mindig, azaz a konvergencia nem biztosított. Ráadásul nem biztos, hogy egy lokális egyensúly jó eredményt ad, a modell hajlamos arra, hogy az adathalmaz egy részének megfelelő képeket generáljon - ha például számjegyek írására szeretnénk megtanítani, a Generátor csak egyfajta számjegy generálásával is átverheti a Diszkriminátort. Emellett a GAN implicit tanulja meg a $p^*(x)$ mögöttes eloszlást, amiről így további információt nem nyerhetünk.

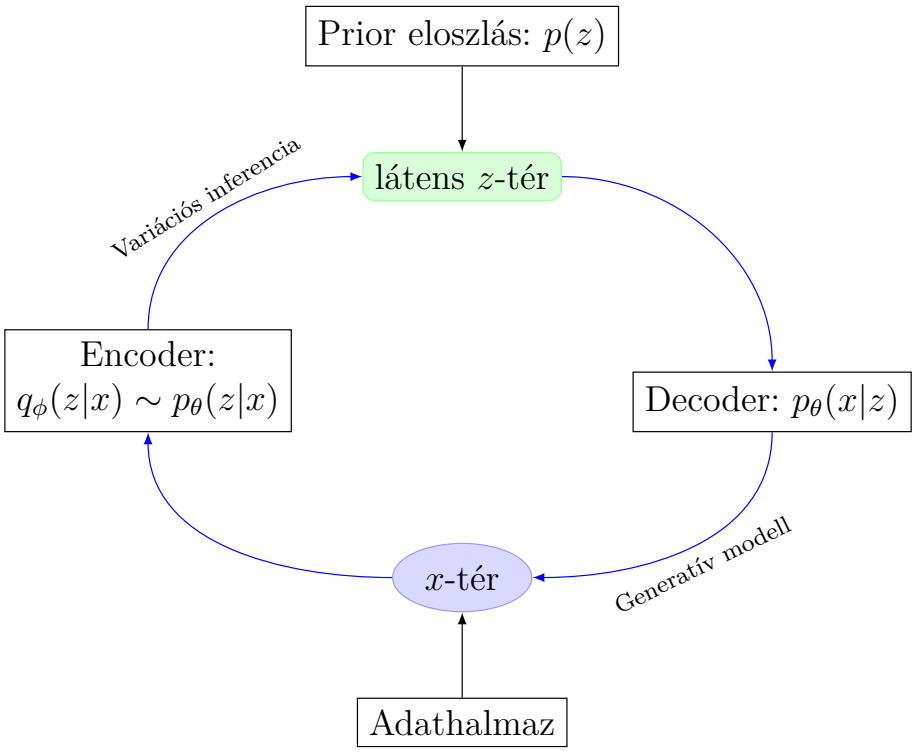
2.3. Variational Autoencoder (VAE)

Egy másik lehetőség az önálló tanulás mentén a *Variational Autoencoder* (VAE), ami az általam épített modellek alapját képezi. Ez a bayesi statisztika eszközeivel tanul látens reprezentációt és eloszlást az adathalmazra nézve. Az alábbiakban a modell alkotói, Kingma és Welling (2019) leírása alapján foglalom össze a VAE tulajdonságait.

2.3.1. Általános leírás

A VAE a bayesi statisztika eszközeivel dolgozik. Most is adott az adathalmaz és az ismeretlen mögöttes eloszlás: $p^*(x)$. A célunk az lesz, hogy egy x adatpontnak egy megfelelő z látens térbeli reprezentációt találjunk. Ehhez $p(z)$ eloszlásra meghatározunk egy prior becslést. A modell két részből fog állni: ez az Encoder és a Decoder. Az Encoder feladata, hogy egy adott x-re előállítsa a látens z megfelelőjét, meghatározva a posterior $p_\theta(z|x)$ feltételes eloszlást. Látni fogjuk, hogy ezt nem tudjuk analitikusan számolni, hanem egy $q_\phi(z|x)$ variációs posteriorral kell közelítsük - ezért ezt a folyamatot *variációs inferenciá*nak hívjuk. A Decoder feladata az, hogy egy látens térbeli z értékből visszaállítsa az eredeti x inputot: $p_\theta(x|z) \sim p^*(x)$. Ez a modell generatív része, hiszen a tanítás után egy tetszőleges z vektorral új képeket is tudunk generálni. A modell általános felépítése a (2.1) ábrán látható. Jellemzően, így esetünknben is, az Encodert és Decodert egy-egy nemlineáris neurális háló

valósítja meg.



2.1. ábra. A VAE általános struktúrája

A cél tehát a $p^*(x)$ eloszlás közelítése, azonban továbbra sem számítunk arra, hogy ez egy explicit paraméterezhető eloszlás lesz. Ezért vezetjük be a látens $p(z)$ *priort* (ami gyakorlatban sokszor egy normál vagy uniform eloszlás). Az eredeti $p^*(x)$ eloszlást ennek függvényében szeretnénk keresni: ehhez elsőként meghatározunk egy összetett f_θ függvényt. Ilyen lehet például egy mély neurális háló. A függvény kimenetét esetlegesen tovább transzformálhatjuk valamilyen g függvénnyel, hogy megkapjuk az x változó közelítését a prior függvényében:

$$x \sim g(f_\theta(z)).$$

Ebből adódik egy együttes eloszlás x-re és z-re: $p_\theta(x, z)$. Ebből könnyen tudunk mintát venni: $z \sim p(z)$ és $x \sim g(f_\theta(z))$ szerint. Viszont az eredeti p^* eloszlás csak x függvénye, ezért p_θ x-re vett marginális eloszlására van szükségünk. Ezt általános esetben integrálással kaphatjuk meg:

$$p_\theta(x) = \int p_\theta(x, z)dz = \int p_\theta(x|z)p(z)dz \tag{2.2}$$

Az így létrejött p_θ eloszlás elég flexibilis tud lenni. Egy egyszerű eset: ha z diszkrét, $p(z|x)$ pedig normál eloszlás, akkor normál eloszlások affin kombinációjáról beszélhetünk. Folytonos z esetén pedig összetettebb eloszlást kaphatunk. A továbbiakhoz idézzük fel Bayes-tételét.

2.1. Tétel (Bayes-tétel)**.** *Legyen X és Y két valószínűségi változó, $p(X)$ és $p(Y)$ az eloszlásuk. Ekkor az egymásra vett feltételes eloszlásokra fennáll a következő összefüggés:*

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)}$$

A mögöttes eloszlás közelítéséhez, $p_\theta(x)$ meghatározásához a (2.2) integrál megoldására van szükségünk. Ez azonban tipikusan lehetetlen, mert nem ismert rá analitikus megoldás, vagy jó közelítés. Ha a 2.1 Bayes-tételt az x és z valószínűségi változókra nézzük, akkor átszorzással kapjuk, hogy:

$$p_\theta(x) = \frac{p_\theta(x|z)p(z)}{p_\theta(z|x)} \tag{2.3}$$

A jobboldal számlálója a fentebb leírtak szerint könnyen számolható, azonban a nevezőben szereplő $p_\theta(z|x)$ posterior szintén ismeretlen. Ezt a problémát hidalja át a VAE, amikor az optimalizáció során variációs közelítést ad a posteriorra.

2.3.2. Veszteségfüggvény

A (2.1) formula szerinti minimalizáláshoz szükségünk van egy megfelelő veszteségfüggvényre, aminek mentén optimalizálni tudunk. Ehhez szükségünk lesz a KL divergencia fogalmára, ami azt határozza meg, hogy egy eloszlás mennyire különböző egy másik, referencia-eloszlástól. A jelölt alap nélküli logaritmusfüggvény a természetes alapú logaritmus.

Definíció (KL divergencia)**.** Legyen p és q folytonos valószínűségi változók sűrűségfüggvénye. Ekkor az eloszlásaik Kullback-Leibler divergenciája a következő integrál értéke:

$$D_{KL}(p\|q) = \int_{\mathbb{R}} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx = \mathbb{E}_p\left[\log\frac{p(x)}{q(x)}\right]$$

A KL divergencia nem távolságfüggvény, ugyanis nem szimmetrikus és nem teljesíti a háromszög-egyenlőtlenséget. A nemnegativitást viszont teljesíti, erre szükségünk is lesz a későbbiekben.

2.2. Állítás. $\forall p, q : D_{KL}(p\|q) \geq 0$

Bizonyítás. Azt fogjuk belátni, hogy $-D_{KL}(p\|q) \leq 0$.

$(*)$ Vegyük észre, hogy $a > 0 \Rightarrow \log a \leq a - 1$ és $\frac{q(x)}{p(x)} > 0$.

$$\begin{aligned}
-D_{KL}(p\|q) &= -\int_{\mathbb{R}} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx \\
&= \int_{\mathbb{R}} p(x) \log\left(\frac{q(x)}{p(x)}\right) dx \\
&\overset{(*)}{\leq} \int_{\mathbb{R}} p(x) \left(\frac{q(x)}{p(x)} - 1\right) dx \\
&= \int_{\mathbb{R}} q(x) - p(x) dx \\
&= 1 - 1 = 0
\end{aligned}$$

mivel $p(x)$ és $q(x)$ sűrűségfüggvények. □

A (2.3.1) részben láttuk, hogy egy olyan modellt szeretnénk készíteni, aminek a $q_\phi(z|x)$ variációs posteriora jól közelíti a valódi és ismeretlen $p_\theta(z|x)$-t. A közelség mértékét KL divergencia segítségével tudjuk számszerűsíteni (az argumentum nélküli q_ϕ illetve p_θ a $q_\phi(z|x)$ és $p_\theta(z|x)$ eloszlásokat rövidítik).

$$D_{KL}(q_\phi\|p_\theta) = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right] = \mathbb{E}_{q_\phi}[\log q_\phi(z|x)] - \mathbb{E}_{q_\phi}[\log p_\theta(z|x)] \tag{2.4}$$

Ahonnan a második tagot továbbírva:

$$\begin{aligned}
\mathbb{E}_{q_\phi}[\log p_\theta(z|x)] &= \mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(z,x)}{p_\theta(x)}\right] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi}[\log p_\theta(x)] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(z,x)] - \int q_\phi(z|x) \log p_\theta(x) dz \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(z,x)] - \log p_\theta(x) \int q_\phi(z|x) dz \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(z,x)] - \log p_\theta(x)
\end{aligned} \tag{2.5}$$

Ezt visszahelyettesítve a (2.4) egyenletbe:

$$\begin{aligned}
D_{KL}(q_\phi\|p_\theta) &= \mathbb{E}_{q_\phi}[\log q_\phi(z|x)] - \mathbb{E}_{q_\phi}[\log p_\theta(z,x)] + \log p_\theta(x) \\
\log p_\theta(x) &= \underbrace{-\mathbb{E}_{q_\phi}[\log q_\phi(z|x)] + \mathbb{E}_{q_\phi}[\log p_\theta(z,x)]}_{\text{ELBO: } L_{\theta,\phi}(x)} + \underbrace{D_{KL}(q_\phi\|p_\theta)}_{\geq 0}
\end{aligned} \tag{2.6}$$

A KL divergencia nemnegativitása miatt tehát:

$$L_{\theta,\phi}(x) = \log p_\theta(x) - D_{KL}(q_\phi \| p_\theta) \leq \log p_\theta(x) \tag{2.7}$$

Vagyis a $p_\theta(z|x)$-re tett $q_\phi(z|x)$ variációs becslés jósága, amit a (2.4) egyenletben szereplő KL divergenciával határozunk meg, az ELBO becslés pontosságát is meghatározza. Minél jobban közelíti $q_\phi(z|x)$ az igazi posteriort, annál jobban közelíti az ELBO a $\log p_\theta(x)$ loglikelihoodot. És *vica versa*, az ELBO maximalizálásával egyrészt növekszik a $p_\theta(x)$ marginális likelihood, azaz jobb lesz a generatív modell, emellett csökken a KL divergencia, vagyis jobb lesz az variációs posterior becslésünk.

Az ELBO-t a (2.6) alak mellett a következő, szemléletesebb módon is felírhatjuk:

$$\begin{aligned} L_{\theta,\phi}(x) &= \mathbb{E}_{q_\phi}[\log p_\theta(z,x)] - \mathbb{E}_{q_\phi}[\log q_\phi(z|x)] \\ &= \mathbb{E}_{q_\phi}[\log p_\theta(x|z)] + \mathbb{E}_{q_\phi}[\log p_\theta(z)] - \mathbb{E}_{q_\phi}[\log q_\phi(z|x)] \\ &= \underbrace{\mathbb{E}_{q_\phi}[\log p_\theta(x|z)]}_{\text{rekonstrukció jósága}} - \underbrace{\mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(z|x)}{p_\theta(z)}\right]}_{\substack{\text{a becsült posterior és a prior} \\ \text{KL divergenciája}}} \end{aligned} \tag{2.8}$$

Két dolgot szeretnénk tehát elérni. Az egyik, hogy minél pontosabb legyen a rekonstrukció. A másik, hogy a posterior hasonló legyen a prior eloszláshoz - ezt hívjuk regularizációs tagnak. Ez biztosítja, hogy a látens tér elrendezése értelmes reprezentációt mutat, azaz (normál prior esetén) az elkódolt értékek az origóhoz közel helyezkednek el, és a különböző pontok közötti átmenet folytonos. Ellenkező esetben ugyanis a látens tér különböző részeiben képezhetné le a modell különböző bemeneteket, így nem alakulna ki valódi információt hordozó látens tér.

2.3.3. Optimalizálás

Egy jó modell megalkotásához tehát az ELBO-t kell maximalizálnunk egy optimalizációs algoritmussal. Egy mély neurális hálót fogunk használni a variációs inferencia alapjaként, ennek tanítása pedig a gradiens ereszkedés módszerével (*gradient descent*) történhet, az ELBO-t használva veszteségfüggvényként. A gradiens leszállás során a következőképpen frissítjük a paramétereket az esetünkben:

$$(\phi, \theta) = (\phi, \theta) - \lambda \cdot \nabla_{\phi,\theta} L_{\phi,\theta}(x)$$

Ahol ϕ és θ az Encoder illetve Decoder paraméterei, λ a tanulási ráta. A veszteségfüggvényünk minimalizálása céljából kiszámoljuk a gradienst, és ennek egy λ skalárszorosával változtatunk a paramétereinken, amik így a lokális optimum irányába fognak változni. Ezt a paraméterfrissítést a *minibatch* módszerrel fogjuk végrehajtani, azaz meghatározott számú adatpontot adunk a modellnek, mielőtt ezekből

kiszámolnánk a gradienst. Így stabilabb lesz a konvergencia, mintha minden adatpont után frissítenénk (Ruder, 2016), illetve lehetőségünk nyílik Monte Carlo jellegű becslések használatára is.

A diszkriminatív modellek egyszerű veszteségfüggvényeinél valamivel nehezebb helyzetben vagyunk, ugyanis az ELBO-ban szereplő várható értéket nem tudjuk analitikusan kiszámolni. A generatív modell θ paramétereire nézve az ELBO (2.6) alakjából kaphatunk torzítatlan becslést:

$$\begin{aligned}
\nabla_\theta L_{\phi,\theta}(x) &= \nabla_\theta \left[\mathbb{E}_{q_\phi}[\log p_\theta(x,z)] - \mathbb{E}_{q_\phi}\left[\log q_\theta(z|x)\right]\right] && \text{(2.9a)}\\
&= \mathbb{E}_{q_\phi}\left[\nabla_\theta[\log p_\theta(x,z) - \log q_\phi(z|x)]\right] && \text{(2.9b)}\\
&\simeq \nabla_\theta\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right] && \text{(2.9c)}\\
&= \nabla_\theta[\log p_\theta(x,z)] && \text{(2.9d)}
\end{aligned}$$

Ahol (2.9b)-nél azért cserélhettük fel a várható értéket és a gradienst, mert különböző változókra vonatkoznak. (2.9c)-nél egy Monte Carlo becslést végzünk: a *minibatch* miatt torzítatlanul közelítjük a várható értéket. (2.9d)-nél pedig felhasználjuk, hogy a második tag θ-ra vett deriváltja 0, mert tőle független.

Az Encoder ϕ paramétere esetében nem cserélhetjük fel a várható értéket és a gradienst, ezért itt más módszert kell használjunk. A probléma kiküszöbölésére változócserét fogunk végezni (*reparameterization trick*). Elsőként legyen egy ε valószínűségi változónk, aminek eloszlása független x-től és ϕ-től. Ekkor a $z \sim q_\theta(z|x)$ valószínűségi változót kifejezhetjük a következőképpen:

$$z = g(\varepsilon, \phi, x)$$

egy megfelelő g függvényre. Így a gradiens és a várható érték operátorok kommutatívak lesznek ε függetlensége miatt, és Monte Carlo becsléssel:

$$\begin{aligned}
\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[f(z)] &= \nabla_\phi \mathbb{E}_{p(\varepsilon)}[f(z)]\\
&= \mathbb{E}_{p(\varepsilon)}[\nabla_\phi[f(z)]\\
&\simeq \nabla_\phi f(z)
\end{aligned} \qquad (2.10)$$

Ezzel kiegészítve a (2.9) számolást differenciálhatjuk a teljes reparametrizált veszteségfüggvényünket:

$$\begin{aligned}
\nabla_{\phi,\theta} L_{\phi,\theta}(x) &= \nabla_{\phi,\theta}\left[\mathbb{E}_{q_\phi}[\log p_\theta(x,z)] - \mathbb{E}_{q_\phi}\left[\log q_\theta(z|x)\right]\right]\\
&= \nabla_{\phi,\theta}\left[\mathbb{E}_{p_\varepsilon}[\log p_\theta(x,z)] - \mathbb{E}_{p_\varepsilon}\left[\log q_\theta(z|x)\right]\right]\\
&= \mathbb{E}_{p_\varepsilon}\nabla_{\phi,\theta}[\log p_\theta(x,z) - \log q_\theta(z|x)]\\
&\simeq \nabla_{\phi,\theta}[\log p_\theta(x,z) - \log q_\theta(z|x)]\\
&= \nabla_\theta \log p_\theta(x,z) - \nabla_\phi \log q_\theta(z|x)\\
&= \nabla_{\phi,\theta}\tilde{L}_{\phi,\theta}(x)
\end{aligned} \qquad (2.11)$$

ahol az utolsó sorban $\tilde{L}$ a gradiens érték becslését jelöli. Így pedig a megfelelő programozási keretrendszerben már könnyen elvégezhető a differenciálás, a kapott becslésünk pedig torzítatlan, hiszen:

$$\begin{aligned}\mathbb{E}_{p(\varepsilon)}\left[\nabla_{\phi,\theta}\tilde{L}_{\phi,\theta}(x,\varepsilon)\right] &= \mathbb{E}_{p(\varepsilon)}(\nabla_{\phi,\theta}[\log p_\theta(x,z) - \log q_\theta(z|x))] \\ &= \nabla_{\phi,\theta}[\mathbb{E}_{p(\varepsilon)}(\log p_\theta(x,z) - \log q_\theta(z|x))] \\ &= \nabla_{\phi,\theta}L_{\phi,\theta}(x)\end{aligned} \tag{2.12}$$

Most már minden készen áll a modell létrehozásához és tanításához. A tanítás menetét leírtam fentebb, a pontos algoritmus az eredeti leírásban található (Kingma & Welling, 2014, p. 21).

3. Gaussian Scale Mixtures

A modellek bemutatása előtt a természetes képek statisztikai tulajdonságairól írok röviden. Olshausen és Field (1996) vizsgálják azt, hogy egy természetes képet hogyan lehet hatékonyan elkódolni. A szerzők célja az, hogy egy $n \times n$ pixeles képet, amit az $I(x, y)$ függvénnyel reprezentálunk, felbontsunk (nem feltétlenül ortogonális) bázisfüggvények lineáris kombinációjára:

$$I(x,y) = \sum_i a_i \phi_i(x,y)$$

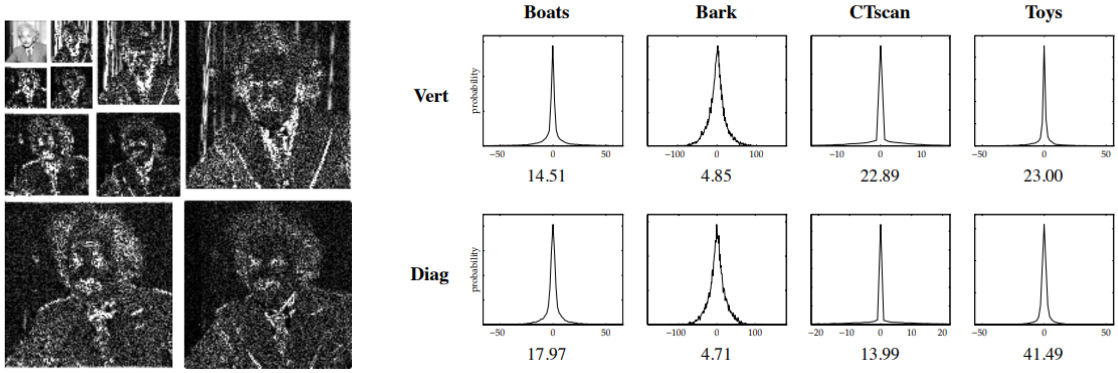
ahol $\phi_i \in \Phi$ pixeltéren értelmezett függvény, amiből n^2 darabot választunk. A cél az, hogy olyan bázisfüggvényeket válasszunk, amik egyrészt egy teljes ϕ bázisát alkotják a képek terének, másrészt az a_i együtthatók minél inkább függetlenek legyenek természetes képek egy halmazára.

A szerzők azzal a feltételezéssel élnek, hogy a természetes képek ritka struktúrával (*sparse sturcture*) rendelkeznek, vagyis egy nagy méretű Φ halmazból kevés elemmel leírhatunk egy képet. Ez azt jelenti, hogy az együtthatók eloszlása unimodális és nulla körüli. Egy ϕ bázis jóságának értékelésére az ELBO-hoz hasonló veszteségfüggvényt választanak a szerzők: az $E = [\text{információ-megőrzés}] + \lambda \cdot [a_i \text{ ritkasága}]$ összefüggés minimalizálása a cél, ahol $\lambda \in \mathbb{R}^+$ skálázza a második tagot. Eredményként azt találják a szerzők, hogy a természetes képeken tanított modell optimális bázisfüggvényei a *wavelet* transzformációnak megfelelő struktúrát mutatnak. Ez azért is érdekes, mert az emlősök elsődleges látókérgében a neuronok receptív mezőire igaz az, hogy helyileg lokalizáltak, orientáltak és egy adott frekvenciatartományra érzékenyek - ezek a tulajdonságok pedig a wavelet transzformációra is igazak.

A *wavelet* transzformációt pedig ezek után a következőképpen pontosíthatjuk. Legyen $\psi \in L^2(\mathbb{R})$, azaz $\int_{\mathbb{R}} \psi^2(x)\, dx < \infty$, továbbá legyen $\int_R \psi(t)\, dt = 0$. ψ ortonormált *wavelet*, ha teljes $\{\psi_{jk} : j, k \in \mathbb{Z}\}$ ortonormált bázis származtatható belőle az

$L^2(\mathbb{R})$ Hilbert-térre. Ekkor $\forall f \in L^2(\mathbb{R})$ felírható $f(x) = \sum_{j,k=-\infty}^{\infty} c_{jk}\psi_{jk}(x)$ alakban, a megfelelő c_{jk} együtthatókra, amit f wavelet-sorba fejtésének nevezünk (Chun-Lin, 2010). Chun-Lin munkájának 31. oldalán látható néhány példa a waveletekre. A mostani jelentéskörnyezetben, a képek terén a waveletek értelmezési tartománya nem a teljes valós halmaz lesz, hanem egy kép $(x, y) \in \mathbb{R}^2$ síkrészletén vesszük őket, a pixelértékekre diszkretizálva. A képeken egy wavelet-bázist tehát n^2 wavelet fog kiadni, amik skálában és orientációban különböznek, ahol a skála a szélességet határozza meg, az orientáció a térbeli elhelyezkedést és irányt (képek és részletes leírás Olshausen és Field (1996) munkájában).

Ezt az eredményt tovább kutatva Wainwright és Simoncelli (1999) foglalkozik természetes képek wavelet transzformáció utáni statisztikájának leírásával. Analízisük szerint a képek *wavelet* bázis szerinti reprezentációi két tekintetben is nem normál eloszlást követnek. Az egyik a korábban említett ritka struktúra: egy kép elkódolásakor az együtthatók legtöbbje 0 körüli és kevés lesz nagy amplitúdójú. A (3.1a) képen látszanak egy képkódolás együtthatóinak abszolútértékei három különböző skála és orientáció szerinti waveletre, láthatóan ritka eloszlással, sok sötét, azaz 0 közeli értékkel. Ezt kvantitatívan láthatjuk a (3.1b) hisztogramon, ahol a középső, 0 érték körül nagyon sok együttható van, két oldalra pedig erősen csökken a számuk. A különböző diagramokon négy különböző dolgot ábrázoló természetes kép (tájkép, fakéreg-szerű textúra, orvosi kép, számítógépes grafika) együtthatóinak eloszlása látható. A felső sorban függőleges, az alsó sorban pedig vízszintes orientációjú, közepes skálájú waveletek együtthatóit láthatjuk. Az eloszlás nyilvánvalóan nem normál, ez tehát az első eltérés.



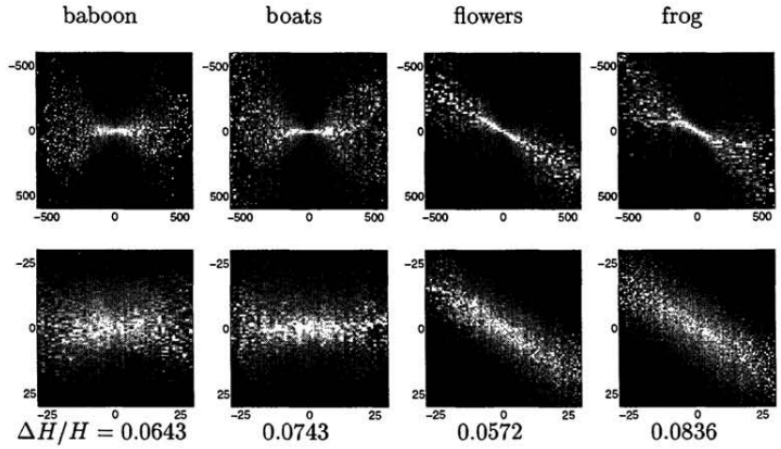
(a) Együtthatók abszolútértékei a kép mentén

(b) Együtthatók eloszlása

3.1. ábra. Természetes képeken vett wavelet transzformációk együtthatóinak tulajdonságai (Buccigrossi & Simoncelli, 1999)

A második, nem normál eloszlás szerinti tulajdonság a wavelet transzformáció

utáni szomszédos pixelek intenzitása (szomszédos ugyanazon wavelet szerinti kódolásban, vagy ugyanaz a pixel hely, csak orientációban, illetve skálában eltérő waveletre). Ha ugyanis egy együttható nulla, akkor a szomszédjai is nagy valószínűséggel lesznek nulla közeliek, azonban egy együttható magas értéke mellett a szomszédoknak lehet nagy, vagy kicsi értéke is, azaz nő a varianciájuk. Ez látható diagramon a (3.2) ábrán, a felső sorban. A vízszintes tengely mentén látszik a független pixel intenzitása (a (3.1b) hisztogramokhoz hasonlóan a középső értékek jelentik a nulla értéket). A függőleges tengely mentén a szomszédok intenzitását látjuk. A szaloncukorra hasonlító alak jelenti a most leírt összefüggést, azaz hogy magas értékek mellett a szomszédok értékei sokfélék lehetnek. Az egyes oszlopok négy különböző természetes képre adott eloszlást jelenítenek meg ugyanazon wavelet szerint. Az alsó sor magyarázatát a szerzők által vázolt megoldás bemutatása után írom le.



3.2. ábra. Kondícionális hisztogram a szomszédos együtthatók értékeire (Wainwright & Simoncelli, 1999)

A fentiek miatt a bemutatott eloszlások modellezésére nem alkalmas normál eloszlások kombinációja. Ehelyett a szerzők bevezetik a *Gaussian Scale Mixtures* fogalmát.

Definíció (*GSM*)**.** Egy Y valószínűségi vektort *Gaussian Scale Mixture*-nek (GSM) hívunk, ha $Y \stackrel{d}{=} zU$, ahol $\stackrel{d}{=}$ eloszlásbeli egyenlőséget jelöl, $z \geq 0$ egy skalár valószínűségi változó, $U \sim \mathcal{N}(0, Q)$ random normál vektor, illetve z és U függetlenek.

Speciálisan, ha z Gamma eloszlás szerinti, akkor a GSM eloszlás jól tudja közelíteni a (3.1b) ábra szerinti együttható eloszlásokat. Amennyiben pedig a kódolás utáni pixel értékek normalizálva vannak z szerint (egy y_0 pixelt y_0/z_0 szerint normalizálva), a normalizált pixel értékek együttes eloszlása normál - ez látható a (3.2)

ábra alsó sorában. Továbbá a normalizált pixel értékek marginális eloszlása is normál eloszlás lesz. A GSM modell további fejlesztésével pedig nem csak az egymáshoz közeli pixelek válnak jól modellezhetővé, hanem a teljes képen vett együtthatók eloszlása is.

A bemutatott elemzések természetes képekre vonatkoznak, ahol a lineáris tulajdonságok (*feature*-ök) teljes bázist alkotnak. Ennek a leképezéséhez a látens teret az eredeti kép méretének megfelelő, n^2 dimenziósra kéne állítsuk, ez azonban idő- és erőforrásigényes. Ezért a dolgozatban a természetes képeknél egyszerűbb adatbázison, a MNIST augmentált változatán tanítom a modelleket, ahol a tulajdonságok tere (*feature space*) néhány dimenzióban leírható. A célja ennek a modellek működésének megértése és előzetes vizsgálata, hogy a természetes képek vizsgálata ezekből a tapasztalatokból kiindulva történhessen. Tehát a cél kontrasztinvariáns reprezentáció létrehozása a MNIST augmentációján, ahonnan tovább lehet lépni a természetes képekre. Ezért motivál a *scale mixtures* modellek megalkotásában a fenti GSM eloszlás, ami egy hatékony eszköz lehet a neurális háló kontrasztinvariáns megvalósításában.

4. A modellek bemutatása és kiértékelése

4.1. A c-MNIST adathalmaz

A modellt a MNIST adathalmaz egy augmentált változatán tanítom és tesztelem. A MNIST 28x28 pixelből álló, kézzel írott számjegyeket tartalmaz, szürkeárnyalatos formában. Minden képet egy $M \in \mathbb{R}^{28\times 28}$ mátrix reprezentál, amiben minden pixelértékre: $m_{i,j} \in [0, 1]$. Az adathalmaz tanításra szánt része 60000, a tesztrész 10000 képet tartalmaz. Meg kell jegyeznem, hogy ez nem egy nehezen tanulható adathalmaz: régóta léteznek rá szinte tökéletes klasszifikáló algoritmusok, illetve egy szimpla lineáris klasszifikáció is 12%-os hibával megoldja (LeCun, Bottou, Bengio, & Haffner, 1998; LeCun, Cortes, & Burges, n.d.). Az adathalmaz tehát kvázi lineáris.

A modelleket a MNIST egy bővített változatára tanítom. Az eredeti képeket kontrasztált változatokkal egészítem ki: egy eredeti x kép helyett bekerül $c \cdot x$ ($\forall c \in \mathcal{C}$ skalár), ahol $\mathcal{C}$ a kontraszt értékeket tartalmazó halmaz. A vizsgálatom során a $\mathcal{C} = \{0.2, 0.5, 0.8, 1\}$ kontraszt értékekkel augmentálom az eredeti adathalmazt (vagyis az eredeti képek mellett három kontrasztos változatuk is bekerül). Így a tanító adathalmaz összesen 240000, a teszthalmaz 60000 képet tartalmaz. Ezen az augmentált c-MNIST adathalmazon tanítom a modelleket. A modellek felépítését és tanítását a Google Colab keretrendszerében, a *pytorch* deep learning csomag felhasználásával végeztem.

4.2. Standard VAE

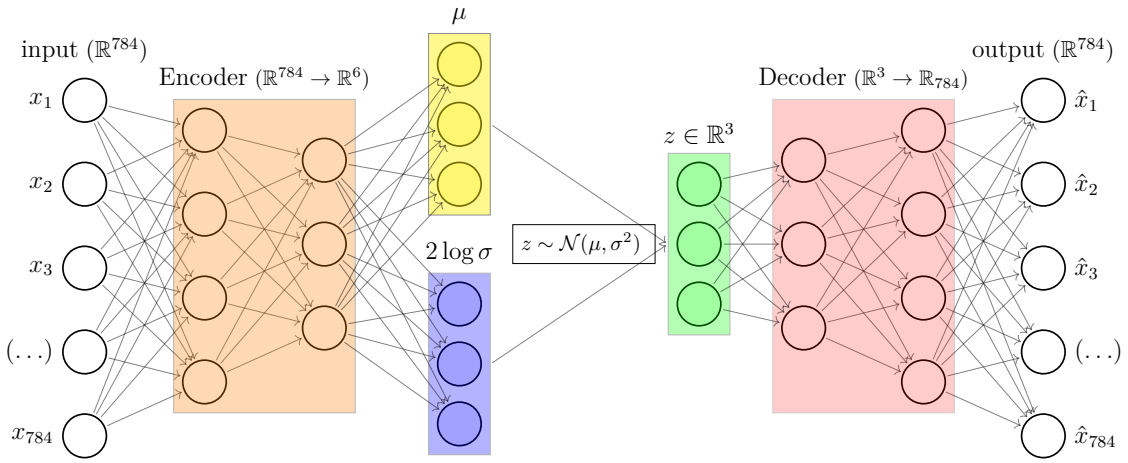
A modell felépítése Az első modell egy standard Variational Autoencoder megvalósítása a c-MNIST adathalmazon. A modell felépítése a következő (ahogyan a (4.1) ábrán is látszik):

- INPUT: $x \in \mathbb{R}^{784}$
- Encoder: $\mathbb{R}^{784} \to \mathbb{R}^6$

  3 rejtett réteg ReLU aktivációs függvénnyel: $784 \to 256 \to 32 \to 6$

  output: $(\mu, 2\log\sigma)$, ahol $\mu, \sigma \in \mathbb{R}^3$ a posterior normál eloszlás paraméterei
- Mintavétel: $z \sim \mathcal{N}(\mu, \sigma^2)$
- Decoder: $\mathbb{R}^3 \to \mathbb{R}^{784}$

  3 rejtett réteg ReLU aktivációs függvénnyel: $3 \to 32 \to 256 \to 784$
- OUTPUT: rekonstruált $\hat{x} \in \mathbb{R}^{784}$



4.1. ábra. A Standard VAE felépítése

A neurális háló Encoder része inputként egy $x \in$ c-MNIST 28x28-as szürkeárnyalatos képet kap meg egy 784 dimenziós vektorrá kilapított formában. Az Encoder kimenetként az adott x kép alapján kapott $q_\phi(z|x)$ variációs posterior eloszlás paramétereit adja (ez közelíti a valódi $p_\theta(z|x)$ posterior eloszlást). Ebben a modellben a látens tér 3 dimenziós, a prior normál eloszlást követ: $p_\theta(z) \sim \mathcal{N}(0, I)$. A variancia nemnegativitása miatt a megfelelő paraméterekre pozitív értéket várunk, ezt azonban nem tudjuk beleépíteni a neurális hálóba. Emiatt a variancia logaritmusa

lesz az Encoder kimenete: $\log \sigma^2 = 2 \log \sigma$. Ebből könnyen kiszámolhatjuk a normál eloszlás varianciáját: $\exp(0.5 \cdot 2 \log \sigma)$.

A következő, sztochasztikus rétegben mintát veszünk az Encoder által meghatározott paraméterű eloszlásból. A (2.3.3) részben leírtak szerint változócserére van szükségünk (*reparameterization trick*), hogy a deriváltakat megkapjuk, és gradiens módszerrel tudjunk optimalizálni. Ezt a konkrét modellben a *pytorch* distribution csomagjának segítségével valósítottam meg, ami a szükséges módon vesz mintát a megadott eloszlásból (PyTorch, 2023; Dillon és tsai., 2017).

Hiperparaméterek és optimalizálás A hiperparaméterek az összes megalkotott modellben megegyeznek. A látens dimenzió, azaz a z tér háromdimenziós, az MNIST egyszerűsége miatt ugyanis elég lesz kevés látens dimenzió. Ha nagyobb látens teret veszünk, azt láthatjuk, hogy a posterior "összeomlik" pár dimenzió mentén: teljesen megegyezik a priorral, azaz nem hordoz új tulajdonságokat. Ez az alacsony dimenzió továbbá lehetőséget ad arra, hogy a látens tér struktúráját az egyik változó szerinti 2 dimenziós szeleteken láthassuk. A tanítás a gradiens leszállással, azon belül az *Adaptive Moment Estimation* (Adam) módszerrel lett megvalósítva. Ennek hiperparaméterei: batch méret = 128, tanulási ráta = 10^{-3}. Mindkettő egy szokásos érték, és nem találtam náluk jobb eredményt elérő paramétereket. A tanítás 20 *epoch*-on keresztül zajlott, vagyis a tanulás során 20-szor iterált végig a teljes adathalmazon a tanító algoritmus. Ez minden modell esetében elég volt arra, hogy a veszteségfüggvény stabilan megközelítsen egy végső értéket.

Az ELBO megvalósítása Az ELBO-t a korábban látott (2.8) egyenlet szerint valósítjuk meg, vagyis:

$$L_{\theta,\phi}(x) = \underbrace{\mathbb{E}_{q_\phi}[\log p_\theta(x|z)]}_{\text{rekonstrukció jósága}} \quad - \quad \underbrace{\mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(z|x)}{p_\theta(z)}\right]}_{\substack{\text{a becsült posterior és a prior} \\ \text{KL divergenciája}}}$$

A neurális hálók esetében az optimalizálás a legtöbbször minimalizálásként történik, és ez a (2.3.3) részben ismertetett gradiens leszállás módszerre is igaz. Ezért az ELBO maximalizálása helyett a negáltjának minimalizálása lesz a célunk:

$$\max_{\phi,\theta} L_{\phi,\theta}(x) = \min_{\phi,\theta} -L_{\phi,\theta}(x)$$

Az ELBO első tagja a modell rekonstrukciójának pontosságát méri: adott variációs posterior melletti z-re minél nagyobb valószínűséggel szeretnénk az eredeti x-et kapni. A logaritmusfüggvény miatt az eredeti ELBO-ban a várható érték maximuma

0, vagyis a negáltban a 0 felé szeretnénk minimalizálni ezt a tagot. Az implementációban az eredeti x vektort hasonlítjuk össze a rekonstruált $\hat{x}$ vektorral. Erre például a bináris kereszt-entrópiát (*Binary Cross Entropy*, BCE) vagy a négyzetes középérték hibát (*Mean Squared Error*, MSE) használhatjuk, amiket az itt szereplő, $[0, 1]$-beli értékeket tartalmazó vektorokra definiálok.

Definíció. (Bináris kereszt-entrópia) Legyen $x, y \in [0, 1]^N$ vektor. A köztük vett bináris kereszt-entrópia:

$$l_{BCE}(x, y) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log x_i + (1 - y_i) \log(1 - x_i)$$

Definíció. (Négyzetes középérték hiba) Legyen $x, y \in [0, 1]^N$ vektor. A köztük vett négyzetes középérték hiba:

$$l_{MSE}(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2$$

Természetes feltételezni, hogy az eredeti és a rekonstruált kép közötti hiba, azaz a megfigyelési zaj (*observation noise*) normál eloszlású, továbbá a pixelértékek is normál eloszlásúak, ebben az esetben pedig az MSE az adódó választás. A tapasztalataim szerint azonban a BCE használatával jobb eredményeket értek el a modellek. A BCE feltételezése az, hogy a pixelek eloszlása bináris, ekkor működik optimálisan. Ez esetünknben nem igaz, hiszen folytonosak a pixelértékek, viszont az eloszlásuk bimodális, 0 és 1 körüli csúccsal, vagyis a bináris feltételezés pontosabb, mint a normál. A modellek bemutatásakor látni fogjuk, hogy a bináris feltételezés valamelyest elmosódott számalakokat fog eredményezni, ugyanis 0 és 1 közötti pixelértékek esetén a BCE képlete 0.5 felé húz (mivel a másik irányba tévedni nagyobb veszteséget jelentene). Loaiza-Ganem és Cunningham (2019) ezt kijavítva egy BCE variánst alkot, ami folytonos pixel eloszlást feltételez. Ezzel továbbfejleszthető lenne a modell, azért nem valósítottam meg, mert a természetes képek, azaz a cél adathalmaz esetében az MSE módszer feltételezései lesznek természetesek.

A KL divergencia, azaz a regularizációs tag kiszámításához azzal a feltételezéssel élünk, hogy a posterior eloszlás nagyjából egy diagonális kovarianciamátrixú normál eloszlást követ, azaz a posteriorban is függetlenek a különböző dimenziók (lásd Kingma és Welling (2014), p. 5). Ekkor a KL divergenciát két normál eloszlás között kell számoljuk, amire ez analitikusan elvégezhető.

4.1. Állítás. *Legyen $p \sim \mathcal{N}(x; 0, I)$ és $q \sim \mathcal{N}(x; \mu, \Sigma^2)$ k dimenziós normál eloszlás. Ekkor*

$$D_{KL}(q \| p) = \frac{1}{2} \left[\mu^2 + tr\{\Sigma\} - k - \log |\Sigma| \right]$$

Bizonyítás. A sűrűségfüggvények:

$$f_q(x) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

$$f_p(x) = \frac{1}{(2\pi)^{k/2}} \exp\left(-\frac{1}{2}\mathbf{x^2}\right)$$

Ezt behelyettesítve:

$$\begin{aligned}
D_{KL}(q\|p) &= \mathbb{E}_q\left[\log f_q(x) - \log f_p(x)\right] \\
&= \mathbb{E}_q\left[-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu) - \log(2\pi)^{k/2} - \log|\Sigma|^{1/2} + \frac{1}{2}\mathbf{x^2} + \log(2\pi)^{k/2}\right] \\
&= -\frac{1}{2}\log|\Sigma| - \frac{1}{2}\mathbb{E}_q\left[(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right] + \frac{1}{2}\mathbb{E}_q\left[\mathbf{x^2}\right]
\end{aligned}$$

Ahonnan a második tagban szereplő $(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu) \in \mathbb{R}$, vagyis írhatjuk $tr\{(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\}$ alakban, ahol $tr\{\}$ a nyom operátor. Ezen belül pedig kommutatívak a tényezők, vagyis megegyezik a következővel: $tr\{(\mathbf{x}-\mu)^T(\mathbf{x}-\mu)\Sigma^{-1}\}$, ahonnan:

$$\begin{aligned}
\mathbb{E}_q\left[tr\{(\mathbf{x}-\mu)^T(\mathbf{x}-\mu)\Sigma^{-1}\}\right] &= tr\{\mathbb{E}_q\left[(\mathbf{x}-\mu)^T(\mathbf{x}-\mu)\Sigma^{-1}\right]\} \\
&= tr\{\mathbb{E}_q\left[(\mathbf{x}-\mu)^T(\mathbf{x}-\mu)\right]\Sigma^{-1}\} \\
&= tr\{\Sigma\Sigma^{-1}\} \\
&= tr\{I_k\} \\
&= k
\end{aligned}$$

Továbbá a harmadik tagban az egydimenziós $\mathbb{E}(x^2) = Var(x) + \mathbb{E}^2(x)$ összefüggésből:

$$\begin{aligned}
\mathbb{E}_q\left[\mathbf{x^2}\right] &= \mathbb{E}_q\left[\sum_{i=1}^{k} x_i^2\right] \\
&= \sum_{i=1}^{k} \mathbb{E}_q[x_i^2] \\
&= \sum_{i=1}^{k} \left(Var(x_i) + \mu_i^2\right) \\
&= tr\{\Sigma\} + \mu^2
\end{aligned}$$

Ezeket visszahelyettesítve pedig megkapjuk az állításban szereplő kifejezést:

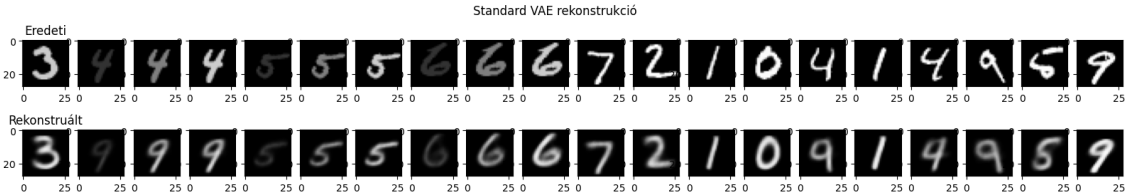
$$D_{KL}(q\|p) = \frac{1}{2}\left[\mu^2 + tr\{\Sigma\} - k - \log|\Sigma|\right]$$

□

Mindezek szerint pedig már minden, a modellnek adott $\mathbf{x}$ adatpontra ki tudjuk számítani az ELBO értékét, ami alapján a (4.2) részben leírtak szerint optimalizálhatunk.

Eredmények A modellek objektív teljesítményét az ELBO teszthalmazra való kiszámításával kaphatjuk meg, ezt a (4.5) részben írom le, minden modellre összesítve.

A modell rekonstrukciójára néhány példa a (4.2) ábrán látható, a felső sorban vannak az eredeti képek, alattuk a rekonstruált párjaik. A rekonstrukció minősége ez alapján jó, a számjegyeket néhány hibával helyesen rekonstruálja, a dőlésszögüket és a négyzeten belüli elhelyezkedésüket is jól állítja vissza. Alacsony kontrasztú képeknél a kontraszttal is pontos, azonban a magas kontrasztú ($c = 1$), azaz világos képek esetében több helyütt sem állítja vissza a teljes kontrasztosságot. A rekonstruált képek valóban kicsit elmosottabb határokkal rendelkeznek.

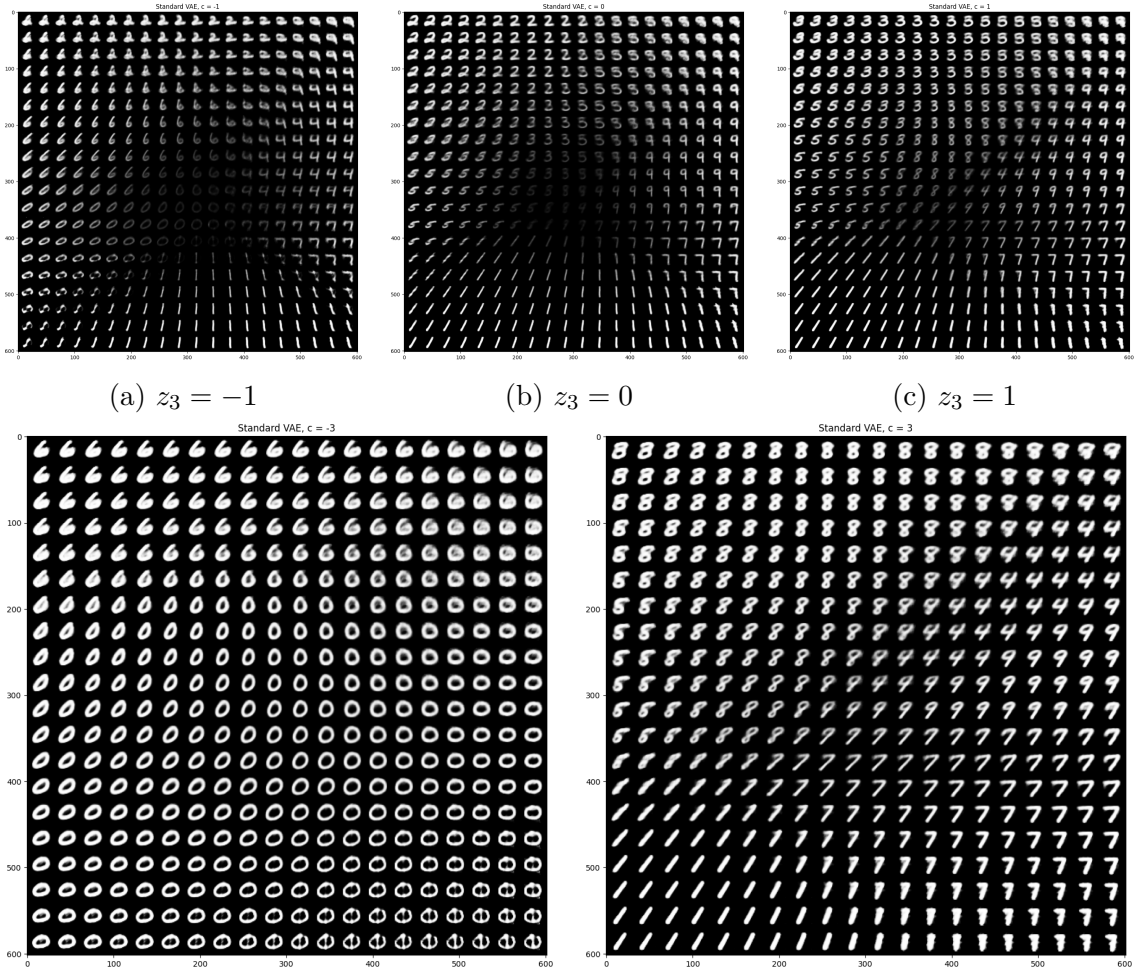


4.2. ábra. Standard VAE rekonstrukció

A modell háromdimenziós látens terét kétdimenziós szeletek mentén tudjuk vizsgálni. A (4.3) ábrán látható a látens tér 5 szelete a $z_3 \in \{-3, -1, 0, 1, 3\}$ értékekre. Az egyes szeleteken a (z_1, z_2) altér értékeinek a Decoder által rekonstruált $\hat{x}$ képei láthatóak. A diagramok mindkét koordinátában a $[-3, 3]$ intervallumot jelenítik meg, 20 ekvidisztáns rácsponton. Tehát az így kapott 400 darab $z \in \mathbb{R}^3$ látens koordinátatrió rekonstrukcióját láthatjuk egy-egy ábrán.

Az ábrákon megfigyelhetjük az általános VAE alaptulajdonságait. A regularizációs tagnak köszönhetően a látens tér elemeinek rekonstrukciója folytonosan változik, azaz közeli látens vektorokból hasonló képet állít elő a Decoder. A rekonstruált számjegyek közti átmenet továbbá értelmes, azaz a látens tér pontjainak túlnyomó többségéből értelmes kép rekonstruálódik. A (4.3d) kép jobb alsó sarkában a kevés kivétel egyik csoportját láthatjuk. Vegyük észre azt is, hogy bár a tanítás során négy különböző értéket felvevő, diszkrét kontraszt változót használtunk, a reprezentáció az ezen kontraszt értékek közötti átmenetben is folytonos.

A kontraszt elkódolását vizsgálva érdekes mintázatot láthatunk. A képek alapján úgy tűnik, hogy a kontraszt az origóban a legkisebb, és az origóból kiinduló félegyenesek mentén növekszik. Ahogyan a (4.4) ábra példáján is láthatjuk, ez valóban így van, sőt úgy látszik, hogy a félegyenesek mentén a számalak is kevéssé változik (a



(a) $z_3 = -1$ (b) $z_3 = 0$ (c) $z_3 = 1$

(d) $z_3 = -3$ (e) $z_3 = 3$

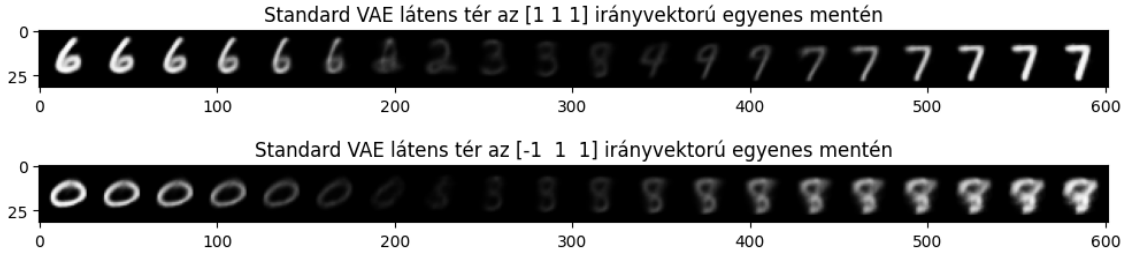
4.3. ábra. Standard VAE látens terének szeletei z_3 mentén

nagyon kicsi kontrasztú részt kivéve). Az ábra két végpontja egyaránt 3 egység távolságra van az origótól, a számjegyek a két végpont közti 20 ekvidisztáns rácspont rekonstrukciói.

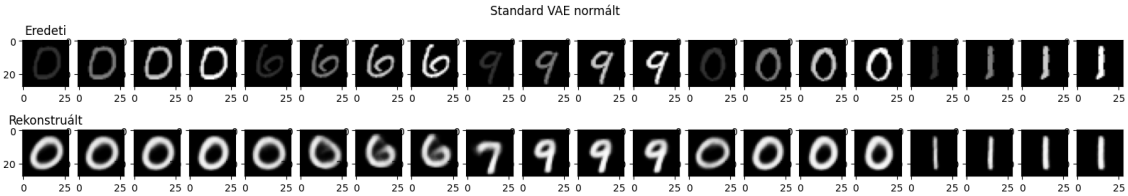
Azt, hogy különböző kontrasztra csak az origótól való távolság változik, a számalaknak megfelelő irányvektor pedig hasonló marad, a (4.5) ábra szemlélteti. Itt a látens tér (z_1, z_2, z_3) kimenetét normáltam, azaz a Decoder a

$$\frac{z}{\|z\|} = \frac{(z_1, z_2, z_3)}{\sqrt{(z_1 + z_2 + z_3)}}$$

inputot kapta. Ez tehát egyedül z iránya, 1 hosszúságra normálva. Az ábrán azt láthatjuk, hogy ugyanazon számalak különböző kontrasztú változatai valóban szinte ugyanolyan irányban kódolódnak el. Eltérést alacsony kontraszt esetén tapasztalunk, ahol nagyobb a modell bizonytalansága a szűkösebb információ miatt.



4.4. ábra. Standard VAE látens tér origón átmenő egyenesek mentén



4.5. ábra. Standard VAE távolságnormált irányvektorok

A (4.3) és (4.4) ábrák alapján tehát az látszik, hogy a modell látens tere radiális szerkezetű. Ennek magyarázatához idézzük fel, hogy a kontraszt változónk egy zárt intervallumon van ($c \in [0, 1]$). A modell feladata (a számjegyek elkódolása mellett), hogy ezt a kontraszt-intervallumot elkódolja az $\mathbb{R}^3$ látens térbe. A regularizációs tag miatt a posterior a priort kell közelítse, a prior normál eloszlás szimmetrikussága miatt pedig akkor tud megvalósulni, ha a kontraszt-intervallumot az egyik féltér irányában kódolja el a modell. Azaz a $[0, 1] \to \mathbb{R}^+$ transzformációval kerülnek a

kontraszt értékek a félegyenesekre, és ez biztosítja a posterior priorhoz való hasonlóságát.

A modell által megtanult radiális reprezentáció kifejezetten izgalmas, illetve jó eredmény az is, hogy a kontrasztot a látens z normája adja meg, a számalaknak megfelelő irányvektor pedig jórészt invariáns a kontrasztra nézve. Ezt úgy szeretnénk továbbfejleszteni, hogy a kontraszt változó értéke egy független dimenzióban jelenjen meg. Ebből a reprezentációból ezt polárkoordinátákra való transzformációval lehetne közelíteni (polárkoordinátás reprezentációval foglalkozik Graving és Couzin (2020) is), viszont ez kívül esik jelen dolgozat keretein. A (4.4) ábrán látottak szerint pedig a kis kontrasztú részen a számjegyek alakja is változik a kontraszt mellett, úgyhogy a függetlenség így sem teljesen adott. A radiális reprezentáció egy jó példa arra, hogy hogyan befolyásolják a modell építése során használt prior feltételezéseink a reprezentációt. A következőkben ezeket a feltételezéseket szeretnénk úgy változtatni, hogy a kontraszt változó egy független euklidészi koordinátán jelenjen meg.

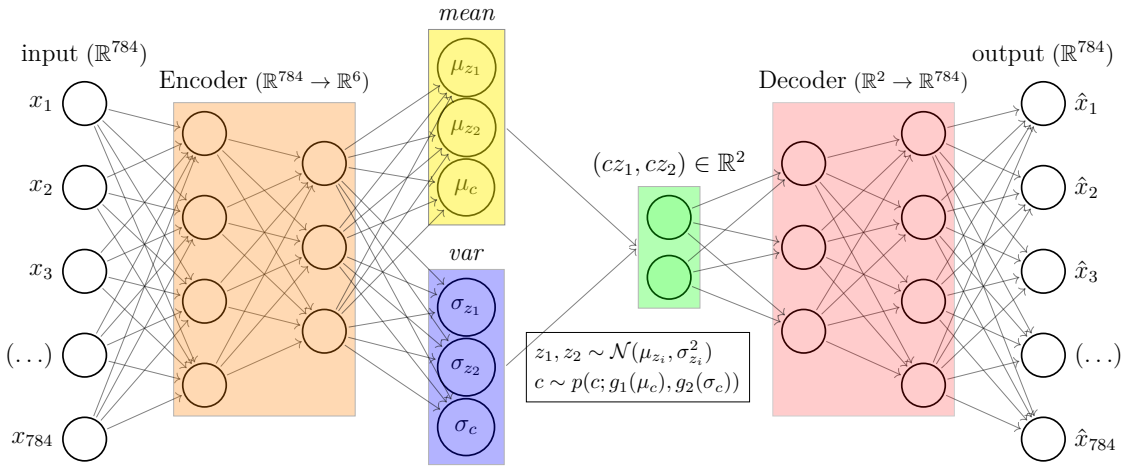
4.3. *Pre Hoc* Scale Mixtures VAE

Az első modellhez képest azt szeretnénk elérni, hogy a kontraszt értéke a látens tér egy független változóján kódolódjon. Az angol nyelvű szakirodalom ezt *disentangled* modellnek hívja, a látens koordináták szétválasztása miatt. Az adathalmaz augmentációjához hasonlóan egy szorzást vezetünk be, amiben az egyik szorzótényezővel a kontraszt értékét szeretnénk kódolni. A szorzás képében bevezetett induktív torzítás inspirációja a Wainwright és Simoncelli (1999) által leírt GSM modell. A VAE struktúra keretei között két, meghatározott eloszlásból származó változó összeszorzásával kapjuk a két eloszlás elegyéből származó *scale mixture* eloszlást. Ezt a szorzást kétféleképpen helyezzük el: a Decoder bemenete előtt, illetve a kimenete után. A bemenet előtti a *pre hoc* (erről lesz szó most), a kimenet utáni a *post hoc* változat ((4.4) rész).

4.3.1. Normál prior

A modell felépítése, változások A modell felépítésében a változás az, hogy a variációs posteriorból vett minta utolsó koordinátájával skalárként megszorozzuk a minta többi részét - azaz a Decoder (z_1, z_2, z_3) input helyett $z_3 \cdot (z_1, z_2)$ inputot kap. Ez a bevezetett induktív torzítás, és a szorzással kapott strukturális változtatáson kívül nem utal más arra, hogy ez a kontrasztot reprezentáló dimenzió. Innentől a z_3 koordinátát c-nek hívjuk, ugyanis ez hivatott kódolni a kontraszt értékét. A látens tér így (z_1, z_2, c) alakú, a Decoder inputja $c \cdot (z_1, z_2)$. c dimenziójára új priort is bevezethetünk, az első modell esetében viszont ez egyelőre marad $c \sim \mathcal{N}(0, 1)$. Ezért a Normál Scale Mixtures név, hiszen egy normál eloszlással skálázzuk az számalakok-

ra vonatkozó (szintén) normál eloszlást. Az Encoder outputja továbbra is μ, illetve $2\log\sigma$ a három dimenzió mentén. A változásokat a (4.6) ábrán is követhetjük. A bekeretezett részben c eloszlása, $p(c)$ most normál, ezen a későbbiekben változtatunk, ezért az absztraktabb leírás.



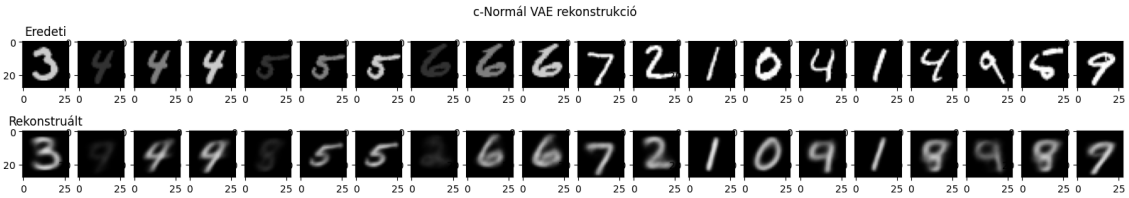
4.6. ábra. A *pre hoc* SMVAE általános felépítése

A szorzás bevezetése kapcsán egy észrevételt kell tegyünk. Azt szeretnénk tehát elérni, hogy a látens tér z koordinátái (illetve a posterior eloszlás első két dimenziója) tartalmazzák a számalakra vonatkozó információt, a c koordináta (és a posterior harmadik dimenziója) pedig a kontrasztot. Mivel a Decoder most egy kétdimenziós inputot kap, $\mathbb{R}^2$-be kell minden információt belesűrítenie a számalakról. Ebben a kétdimenziós térben a c-vel való szorzás félegyeneseket hoz létre, amik mentén változik a kontraszt - ugyanis egy adott (z_1, z_2) vektor c-szeresét véve azt szeretnénk, hogy csak a kontraszt változzon. Vagyis a (z_1, z_2) térben egy már látott radiális elrendezés lenne optimális: origóból kiinduló egyenesek mentén csak a kontraszt változik, a számalak nem, azaz minden irányvektor egy számalakot jelöl.

Vegyük észre, hogy a szorzás miatt valójában nem független a három koordináta, például az $(1, 1, 0.5)$ és a $(2, 2, 0.25)$ látens hármasok ugyanazt a $c \cdot z$ értéket adják: $(0.5, 0.5)$. Ez azt jelenti, hogy egy nagyítás bekerül a rendszerbe: amilyen szám a $c = 0.5$ szeleten az $(1, 1)$ pontról rekonstruálódik, az a $c = 0.25$ szeleten a $(2, 2)$ ponton lesz. c további csökkentésével ez a pont egyre távolabbra kerül, és mivel ez minden pontra igaz, ezért eredményez nagyítást a folyamat, amit a látens téren is látni fogunk. A nagyítás miatt ellentett c értékekre a látens tér egy pontja meg fog egyezni a középpontra vett tükörképével, azaz elég lesz a c szerinti egyik félteret vizsgáljuk.

A modell hiperparaméterei megegyeznek az eredetiével ((4.2)).

Eredmények A modell rekonstrukciója a (4.7) ábrán látható. A számalakokba több hiba csúszik, a dőlésszög és elhelyezkedés viszont pontos. A kontraszt megjelenítése sem fejlődött, a magas kontrasztot nem tudja visszaadni, többi esetben nagyjából pontos. A számjegyek határai elmosottabb határokkal rendelkeznek, mint az előző modell esetében ((4.2) ábra).



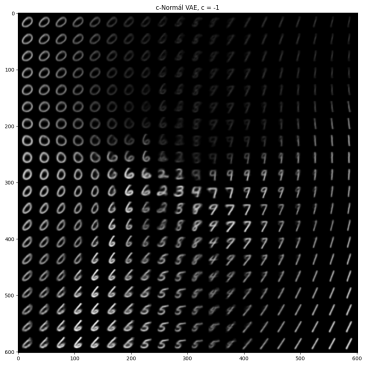
4.7. ábra. Normál VAE rekonstrukció

A látens tér c menti szeleteit a (4.8) ábrán láthatjuk. Ahogyan fentebb írtam, ellentett c értékekre a két tér elrendezése középpontos tükörképe egymásnak, így elég csak c nemnegatív értékeit vizsgálni. A (4.8a) és (4.8c) ábrákon megfigyelhető ez a középpontos szimmetria. A $c = 0$ esetben ((4.8b)) a szorzás miatt lesz teljes homogenitás, hiszen bármilyen $(z_1, z_2, 0)$ hármas a $(0, 0)$ inputot adja a Decodernek. A $c = 1, 2, 5$ szeleteken látjuk, hogy a kontraszt változó hatására hogyan változik a látens tér. A $c = 1$ és $c = 2$ képeken megfigyelhetjük a nagyítás jelenségét: a $c = 1$ szelet a $c = 2$ szelet kétszeres nagyítása, azaz a (4.8d) képen a piros keret a $c = 1$ szeletnek felel meg.

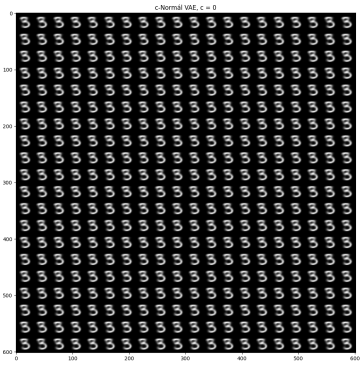
A kontraszt változását figyelve nem nagyon adódik egyértelmű változás a különböző c értékek között. A $c = 1$ nagyítást figyelve nem találunk mintázatot a kontrasztosságban. Azonban a $c = 5$ képpel összehasonlítva látszik, hogy a nagyobb c értékek mentén nagyobb lesz a kontraszt adott (z_1, z_2) értékre (bár a kép bal alsó szeletében megjelenik egy értelmezhetetlen és sötét régió). A c érték tehát tárol kontraszt információt, viszont változtatásával a számjegyek alakja is változik, csak részben teljesül a kívánt tulajdonság, hogy egy origóból kiinduló félegyenes mentén azonos számalakokat találjunk.

Szeretnénk megvizsgálni kvantitatívan is, hogy mennyire teljesül az eredeti célkítűzésünk, a *disentangled*, azaz szétválasztott látens tér. Ehhez újból a teszthalmaz képeit használjuk fel, hogy kereszt-validált eredményt kapjunk. A módszerünk az, hogy a teszthalmaz 10000 képét odaadjuk a modell Encoderének, és elmentjük minden kép kontrasztját, valamint az Encoder által adott posterior eloszlás várható értékét és szórását. A (4.9) ezen értékek átlagát ábrázolja a kontraszt függvényében, a látens tér egyes dimenzióira.

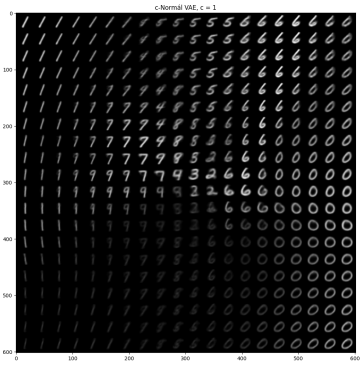
A (4.9a) képen azt láthatjuk, hogy a célunkat egyelőre nem sikerült elérjük. A kontraszt változó hatására z_2 és c alig változik, z_1-gyel viszont szoros kapcsolatban van. A szórás esetében az a várakozásunk, hogy a számalakot hordozó dimenziók va-



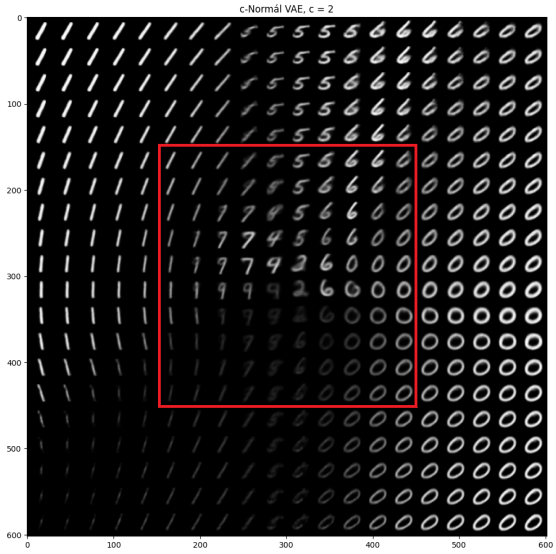
(a) $c = -1$



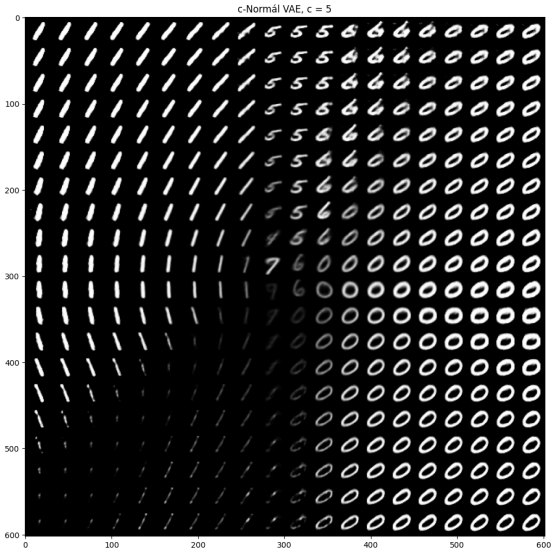
(b) $c = 0$



(c) $c = 1$

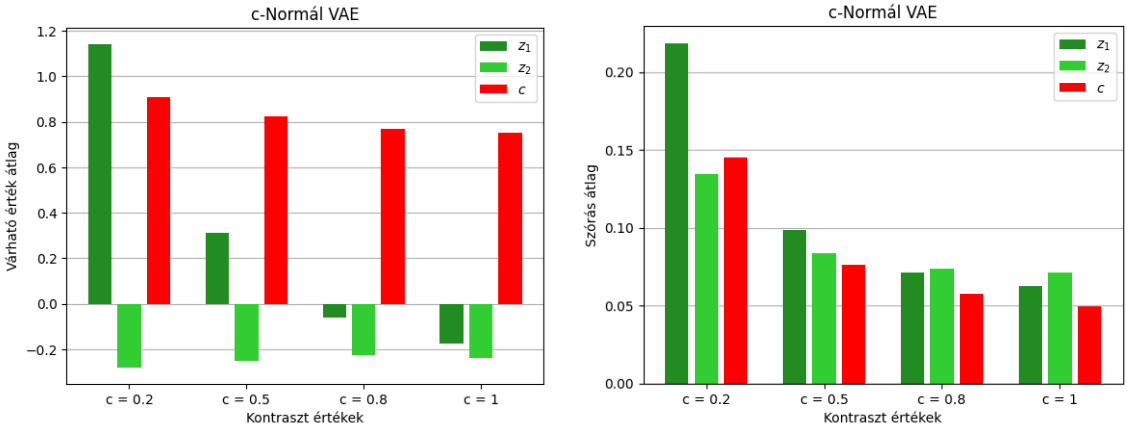


(d) $c = 2$ (piros keretben $c = 1$)



(e) $c = 5$

4.8. ábra. pre hoc Normál SMVAE látens terének szeletei c mentén



(a) Várható érték (b) Szórás

4.9. ábra. Normál VAE posterior várható értékének és szórásának átlaga a kontraszt függvényében

rianciája csökken a kontraszt növekedésével, míg a kontraszt értéket tároló dimenzió varianciája nagyjából azonos marad. Mivel a kontraszt értéket nem sikerült c mentén kódolni, ezért csak általánosságban figyelhetjük meg a variancia csökkenését.

Azt láthatjuk tehát, hogy a modell bizonyos tekintetben jól viselkedik, azonban még sok ponton kell fejleszteni rajta, ugyanis a változók összefüggnek. A következő lépést a c-re vett prior eloszlás megváltozatása fogja jelenteni. A most használt normál eloszlás ugyanis valójában nem modellezi jól c-t, hiszen a kontraszt értékkészlete, a prior normál eloszlás nem felel meg c tartójának. Ezért egy olyan eloszlást lenne érdemes választani, ami csak a pozitív c-t ad. Ilyen lehet az, ha a priorunk gamma eloszlású: $c \sim Gamma(\alpha, \beta)$, a Wainwright és Simoncelli (1999) által használt GSM-hez hasonlóan.

4.3.2. Gamma prior

A modell felépítése A célunk azt megvizsgálni, hogy az előző eredmények javulnake, ha a kontraszt priornak a kontraszt tulajdonságainak jobban megfelelő eloszlást választunk. A gamma eloszlás alkalmas erre a célra, hiszen ha $c \sim Gamma(\alpha, \beta)$, akkor $c \in \mathbb{R}^+$. A gamma eloszlás sűrűségfüggvénye:

$$p(x; \alpha, \beta) = \frac{x^{\alpha-1} e^{-\beta x} \beta^{\alpha}}{\Gamma(\alpha)}$$

α-t *alak*, β-t *skála* paraméternek hívjuk, ugyanis α határozza meg az eloszlás alakját, ami lehet az $\frac{1}{x}$ függvényhez hasonló ($\alpha = 1$), vagy a Poisson-eloszlás alakjára hasonlító ($\alpha = 3$). A skála paraméter egy vízszintes nagyítást végez, de az alakon

nem változtat, ezért ezt $\beta = 1$ értéken fixen hagyjuk, hiszen a modell könnyen át tudja skálázni a kontraszt értékeket, csak az alak számít. Ebből $\alpha = 1$ és $\alpha = 3$ értékeket vizsgáltam.

A modell felépítése a (4.6) ábrához hasonló. A változás tehát a c prior, ami most gamma eloszlás. Az Encoder a $q_\phi(c|x)$ approximációs posteriorra vonatkozóan nem α-t és β-t adja meg, ugyanis ekkor bizonytalanabbnak mutatkozott a tanulás. Ehelyett az eloszlás várható értékének és varianciájának logaritmusát adja meg a modell. Ez a gamma eloszlás esetében

$$\log \mu = \log \frac{\alpha}{\beta} \quad \text{és} \quad \log \sigma^2 = \log \frac{\alpha}{\beta^2}$$

Ezeket könnyen áttranszformálhatjuk a gamma eloszlás α és β paramétereivé:

$$\alpha = \exp \frac{\log \mu^2}{\log \sigma^2} \quad \text{és} \quad \beta = \exp \frac{\log \mu}{\log \sigma^2}$$

Ebből pedig már tudunk mintát venni a (4.2) részben leírtak szerint.

A modell hiperparaméterei megegyeznek az eredetiével ((4.2)).

Az ELBO változása Az ELBO ((2.8)) második, regularizációs tagja fog változni a gamma eloszlás bevétele miatt, ugyanis a KL divergenciát egy együttes normál és gamma eloszlásra kell számoljuk. Ehhez felhasználjuk azt, hogy a prior eloszlás független a három dimenzióban. Továbbá arra a feltételezésre is szükségünk van, hogy a posterior eloszlás dimenziói is függetlenek egymástól. Ez a valóságban nem mindig bizonyul egy helyes feltételezésnek, ami negatívan befolyásolja a modell teljesítményét. Ennek egy lehetséges megoldása a *normalizing flow* technika alkalmazása (Rezende & Mohamed, 2015), azonban ez kívül esik jelen dolgozat keretein. A függetlenség feltételezésére azért van szükségünk, mert ekkor faktorizálhatóak a sűrűségfüggvények: $p_\theta(z, c|x) = p_\theta(z|x)p_\theta(c|x)$ és $q_\phi(z, c|x) = q_\phi(z|x)q_\phi(c|x)$. Ezeket felhasználva kapjuk az alábbi összefüggést.

4.2. Állítás. *Ha feltesszük a posterior eloszlás dimenzióinak függetlenségét, a KL divergenciára:*

$$D_{KL}\Big(q(z, c|x) \,\|\, p(z, c)\Big) = D_{KL}\Big(q(z|x) \,\|\, p(z)\Big) + D_{KL}\Big(q(c|x) \,\|\, p(c)\Big)$$

Bizonyítás.

$$\begin{aligned}
D_{KL}\Big(q(z,c|x)\,\|\,p(z,c)\Big) &= \int q(z,c|x)\log\frac{q(z,c|x)}{p(z,c)}\;dz\;dc \\
&= \int q(z|x)q(c|x)\log\frac{q(z|x)q(c|x)}{p(z)p(c)}\;dz\;dc \\
&= \int q(z|x)q(c|x)\left[\log\frac{q(z|x)}{p(z)}+\log\frac{q(c|x)}{p(c)}\right]dz\;dc \\
&= \int q(z|x)q(c|x)\log\frac{q(z|x)}{p(z)}dz\;dc + \\
&\qquad + \int q(z|x)q(c|x)\log\frac{q(c|x)}{p(c)}dc\;dz \\
&\overset{(*)}{=} \int q(z|x)\log\frac{q(z|x)}{p(z)}\;dz + \int q(c|x)\log\frac{q(c|x)}{p(c)}\;dc \\
&= D_{KL}\Big(q(z|x)\|p(z)\Big) + D_{KL}\Big(q(c|x)\|p(c)\Big)
\end{aligned}$$

ahol a teljes $\mathbb{R}^3$ látens téren integrálunk. A $(*)$ résznél felhasználtuk azt, hogy a sűrűségfüggvény integrálja $\mathbb{R}$-en 1. Továbbá az integrálás sorrendje a Fubini-tétel értelmében itt felcserélhető, mivel korlátos az integrandus L_1 normája. □

A normál priorú dimenziók KL eloszlását már ismerjük ((4.1)), egyedül a gamma prior esetére kell kitérjünk. Az alábbiakban láthatjuk, hogy a gamma eloszlás KL divergenciája is analitikusan számolható.

4.3. Állítás. *Legyen $p \sim Gamma(\alpha_p, \beta_p)$, $q \sim Gamma(\alpha_q, \beta_p)$ egydimenziós gamma eloszlás. Ekkor*

$$D_{KL}(q\|p) = (\alpha_q - \alpha_p)\psi(\alpha_q) - (\beta_q - \beta_p)\frac{\alpha_q}{\beta_q} + \alpha_p\log\frac{\beta_q}{\beta_p} - \log\frac{\Gamma(\alpha_q)}{\Gamma(\alpha_p)}$$

ahol $\psi(z) = \frac{d}{dz}\log\Gamma(z) = \frac{\Gamma'(z)}{\Gamma(z)}$ a digamma függvény.

Bizonyítás. A gamma eloszlás sűrűségfüggvénye:

$$p(x;\alpha_p,\beta_p) = \frac{x^{\alpha_p-1}e^{-\beta_p x}\beta_p^{\alpha_p}}{\Gamma(\alpha_p)}$$

Ezt behelyettesítve:

$$
\begin{aligned}
D_{KL}(q\|p) &= \mathbb{E}_q\left[\log\left(\frac{q(x)}{p(x)}\right)\right] \\
&= \mathbb{E}_q\left[\log\left(\frac{x^{\alpha_q-1}e^{-\beta_q x}\beta_q^{\alpha_q}}{\Gamma(\alpha_q)}\cdot\frac{\Gamma(\alpha_p)}{x^{\alpha_p-1}e^{-\beta_p x}\beta_p^{\alpha_p}}\right)\right] \\
&= \mathbb{E}_q\left[\log\left(x^{\alpha_q-\alpha_p}\cdot e^{x(\beta_p-\beta_q)}\cdot\frac{\beta_q^{\alpha_q}\cdot\Gamma(\alpha_p)}{\beta_p^{\alpha_p}\cdot\Gamma(\alpha_q)}\right)\right] \\
&= \mathbb{E}_q\left[(\alpha_q-\alpha_p)\log x+(\beta_p-\beta_q)x+\alpha_q\log\beta_q-\alpha_p\log\beta_p+\log\frac{\Gamma(\alpha_p)}{\Gamma(\alpha_q)}\right]=
\end{aligned}
$$

Felhasználva a gamma eloszlás várható értékét és a logaritmus várható értékére vonatkozó következő összefüggést:

$$\mathbb{E}_q(x)=\frac{\alpha_q}{\beta_q}\quad\text{és}\quad\mathbb{E}_q(\log x)=\psi(\alpha_q)-\log(\beta_q)$$

az előző egyenletet folytatva kibonthatjuk a várható értéket:

$$=(\alpha_q-\alpha_p)(\psi(\alpha_q)-\log\beta_q)+(\beta_p-\beta_q)\frac{\alpha_q}{\beta_q}+\alpha_q\log\beta_q-\alpha_p\log\beta_p+\log\frac{\Gamma(\alpha_p)}{\Gamma(\alpha_q)}$$

ahonnan a $\log\beta_q$ és $\log\beta_p$ értékeket összevonva kapjuk:

$$=(\alpha_q-\alpha_p)\psi(\alpha_q)-(\beta_q-\beta_p)\frac{\alpha_q}{\beta_q}+\alpha_p\log\frac{\beta_q}{\beta_p}-\log\frac{\Gamma(\alpha_q)}{\Gamma(\alpha_p)}$$

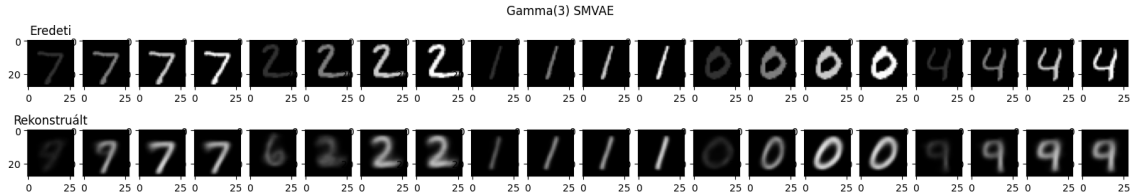
□

A fentiek alapján tehát a regularizációs tag megfelelő átalakításával (ami könnyen leprogramozható) kapjuk az új ELBO-t.

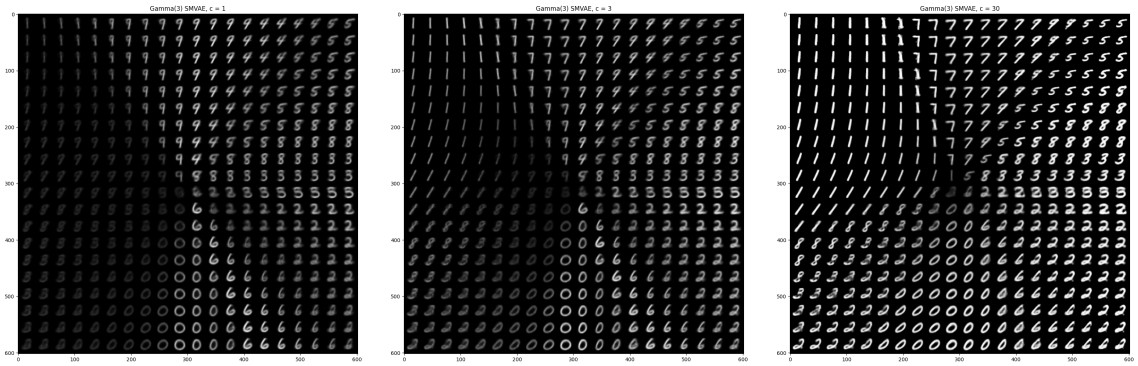
Eredmények Két modellt valósítottam meg a fentiek szerinti konstrukcióban, $Gamma(1,1)$ és $Gamma(3,1)$ priorral. Ezek hasonlóan viselkedtek, ezért ahol nincsen különbség közöttük, csak az egyik eredményeit mutatom ábrán.

A rekonstrukció javult az előző modellhez képest, a kontrasztot jól tanulja, bár a magas kontraszt nem tökéletes. A határok megint elég elmosottak, a számalakok nagyjából helyesek, kis kontraszt esetén jobban hajlamos tévedni ((4.10) ábra).

A Gamma(3) modell látens terét a (4.11) ábrán láthatjuk (a Gamma(1) ehhez hasonló struktúrát mutat). Ellentett c értékekre itt is szimmetrikus a két féltér, $c=0$-ban pedig ugyanazt a homogenitást látjuk, ezért itt már csak pozitív c értékeket mutatok. A nagyítás jelenségét itt is megfigyelhetjük az ábrák között. A kontraszt



4.10. ábra. Pre hoc Gamma SMVAE rekonstrukció



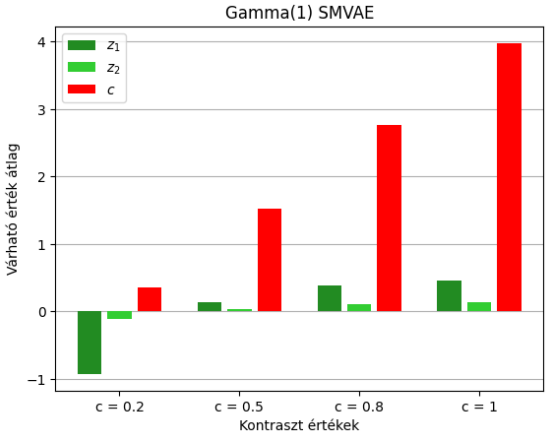
(a) $c = 1$ (b) $c = 3$ (c) $c = 30$

4.11. ábra. pre hoc Gamma SMVAE látens terének szeletei c mentén

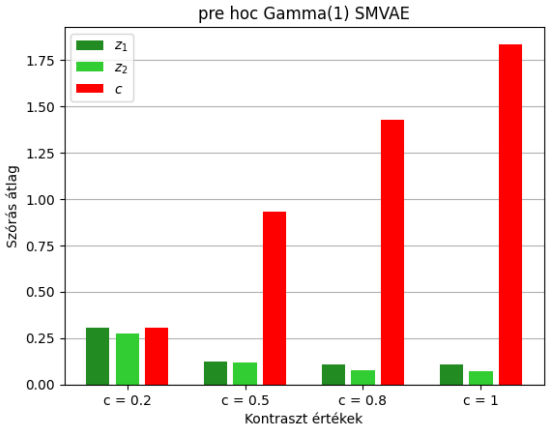
szépen változik c növekedésével, bár elég nagy c értéket kell venni ahhoz, hogy igazán kontrasztos látens teret kapjunk. Mindenesetre egyértelműbb a kontraszt változása, mint az előző modell esetében.

A látens tér kvalitatív megfigyelése alapján tett megfigyelést a hisztogramok is megerősítik. A (4.12a) és (4.12c) ábrákon egyértelműen látszik az összefüggés a kontraszt és a c várható értéke között. Fontos azonban észrevenni, hogy a z dimenziók sem függetlenek a kontraszttól, mindkét ábrán lineárisan változnak a kontraszt hatására, az egyik dimenzió mentén erősebben. A gamma várható értékének nagyságrendje miatt pedig elsőre kisebbnek látszik a változás mértéke: a (4.12a) ábrán például a z_1 dimenzió várható értéke -1 körültől majdnem 0.5-ig nő, ami egy 0 várható értékű normál prior esetén elég nagy változás. A szórás esetében nem teljesül az előzetes várakozásunk, hogy a c dimenzió mentén független a szórás a kontraszttól. Ez valószínűleg annak tudható be, hogy a gamma eloszlás esetében nem független a szórás a várható értéktől. A z dimenziók szórása alacsony, a kis kontraszt sem eredményezett nagyobb szórást.

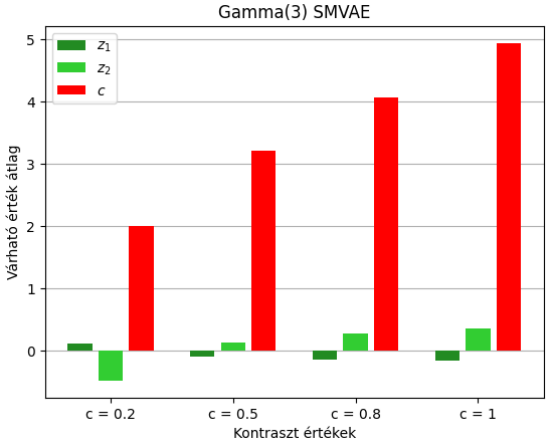
A gamma eloszlású Scale Mixture modell tehát jobban teljesítette a kontraszt független dimenzión való megjelenítésével kapcsolatos célunkat, mint az előző. A kontraszt egyértelműen változik a c dimenzió mentén, azonban nem független a z koordinátáktól. Továbbá azt tapasztaltam, hogy nem stabil a modellek tanulása,



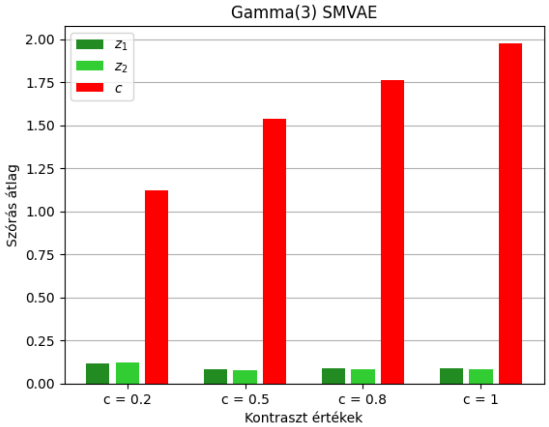
(a) Várható érték, $\alpha = 1$



(b) Szórás, $\alpha = 1$



(c) Várható érték, $\alpha = 3$



(d) Szórás, $\alpha = 3$

4.12. ábra. Pre hoc Gamma SMVAE posterior várható értékének és szórásának átlaga a kontraszt függvényében

egy-egy esetben olyan modell jött létre a tanulás végére, ahol a hisztogramok szerint nem növekedett a c várható érték a kontraszttal. A z koordináták kontraszt függését a Decoder input előtti $c \cdot x$ szorzással együtt járó nagyítás okozhatja, ami lehet, hogy a tanítás stabilitására is rossz hatással van. Az utolsó modellt ezért egy másik megközelítés szerint alkotom meg.

4.4. *Post Hoc* Scale Mixtures VAE

A fent leírt tapasztalatok miatt a (4.3) részben leírt *pre hoc* szorzással ellentétben most *post hoc* szorzással szeretném megragadni a kontrasztinvarianciát. A Decoder a (z_1, z_2) inputot fogja kapni, és a Decoder outputja lesz c-vel megszorozva, azaz $\hat{x} = c \cdot Decoder(z)$ lesz a modell kimenete. Mivel a Decoder nemlineáris, ezért ez más eredményt fog adni, hiszen emiatt $Decoder(c \cdot z) \neq c \cdot Decoder(z)$. Ezzel a megoldással kiküszöböljük a pre hoc szorzás esetében fellépő nagyítást. A kontrasztinvariáns z reprezentáció kialakítása is egyszerűbbé válik, hiszen a látens tér z koordinátáiból kell a számalakot rekonstruálnia a Decodernek, ami után szorozzuk meg ezt a kontraszttól független számalakot a c változóval.

Ilyen felépítéssel három különböző modellt vizsgáltam: c priorra nézve Normál, logNormál és Gamma eloszlásút. A Normál eloszlást elvetettem, mivel a post hoc szorzás miatt csak pozitív c értékekkel szorozhatunk. Erre egy megoldás a logNormál eloszlás használata. Gamma eloszlást is implementáltam $\alpha = 1$, illetve $\alpha = 3$ paraméterekkel, azonban ezek lassan tanultak és nem adtak jó eredményeket, ezért ezeket elhagytam a leírásból.

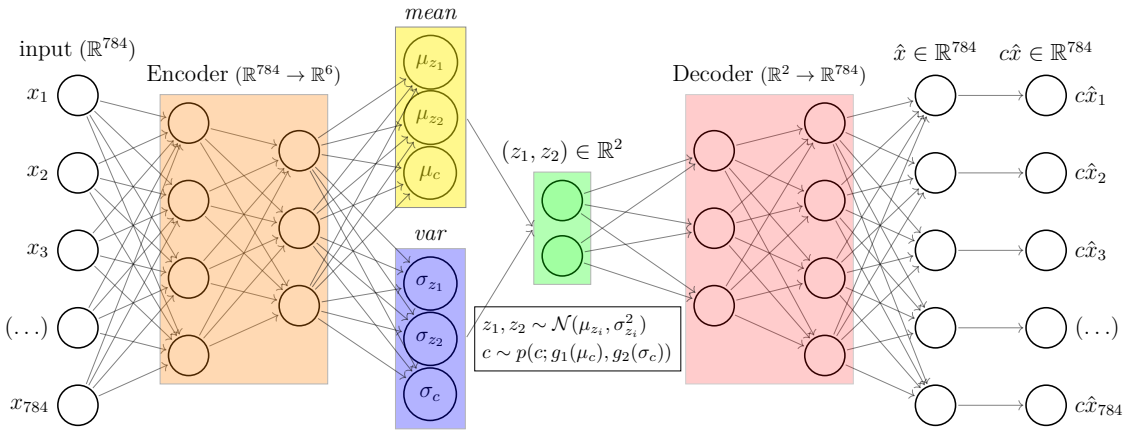
4.4.1. logNormál prior

A modell felépítése Az előző modellhez hasonlóan az Encoder kimenetét (z_1, z_2, c)-vel jelölöm, azonban itt a c-vel való szorzás más helyen történik. A korábbi modellekhez képest tehát a Decoder $z = (z_1, z_2)$ inputot kap, és a $Decoder(z)$ kimenetet szorozza a látens c változó. Jelen modell esetében a c prior logNormál eloszlású, aminek a következő tulajdonságai fontosak. Ha $x \sim logNormal(\mu, \sigma^2)$, akkor $\log x \sim \mathcal{N}(\mu, \sigma^2)$. Ekkor x várható értéke és szórása:

$$\mathbb{E}(x) = \exp\left(\mu + \frac{\sigma^2}{2}\right) \quad \text{és} \quad Var(x) = \exp\left(2\mu + 2\sigma^2\right) - \exp\left(2\mu + \sigma^2\right)$$

Ezek alapján tudjuk majd rekonstruálni a $\log c$-re vonatkozó normál priorból c várható értékét és szórását. Ez a konstrukció pedig a szórással kapcsolatos logaritmusvételhez hasonlóan azt jelenti, hogy az Encoder bármilyen valós számot megadhat kimenetként, az c-re vonatkozóan pozitív értéket fog jelenteni. A modell megvalósításában a logNormál prior annyi változtatást jelent, hogy az Encoder által kapott

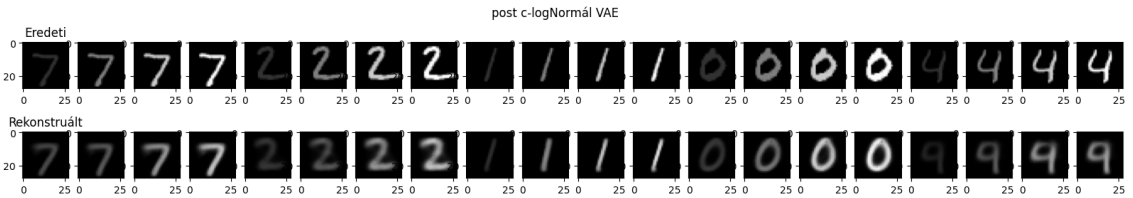
normál eloszlásból vett $\log c$ mintát exponencializálva kell felhasználni a szorzásnál (hiszen ekkor kapjuk meg belőle c-t). Az ELBO-n nem szükséges változtassunk, ugyanis két logNormál eloszlás KL divergenciája megegyezik az ugyanilyen paraméterekkel vett Normál eloszlások KL divergenciájával. A modell felépítését a (4.13) ábrán láthatjuk. A hiperparaméterek megegyeznek a korábbiakkal.



4.13. ábra. A *post hoc* SMVAE általános felépítése

Eredmények A modell rekonstrukciója javult ((4.14), a számalakok kis kontraszt esetén is pontosak. Az ábrán szereplő 4-es alakot a többi modellhez hasonlóan 9-nek nézi. A kontraszt értékek helyesek, a magas kontrasztot nem reprezentálja tökéletesen. Intuíció alapján magabiztosabbnak tűnik a modell, mint a pre hoc verziók.

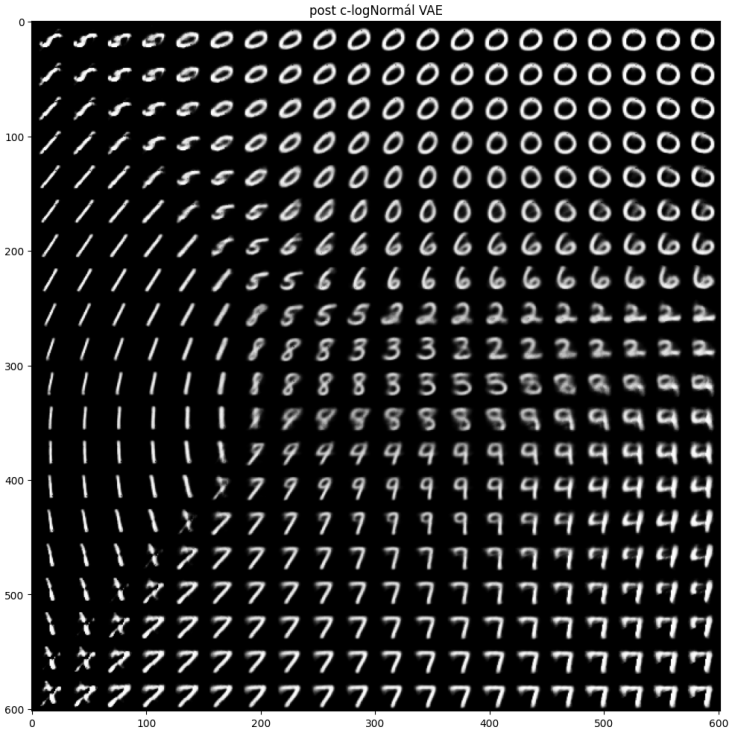
A látens reprezentációt jelenleg a $(z_1, z_2, 1)$ kétdimenziós szeleten tudjuk vizsgálni, ez látszik a (4.15) ábrán. Az ábrán látható számok ugyanis a Decoder rekonstrukciói egy-egy látens térbeli vektor alapján (a korábbiakhoz hasonlóan most is a $(-3, 3)$ intervallumot látjuk z_1 és z_2 szerint egyaránt, 20 ekvidisztáns pontra diszkretizálva). A Decoder pedig csak z inputot kap (c-vel való szorzás nélkül), így a kimenete csak z-től fog függni. Ezt a kimenetet szorozza utólag c, amit csak valódi elkódolt szám esetén kapunk meg, egyedül a látens reprezentációt vizsgálva tehát a



4.14. ábra. Post hoc logNormál SMVAE rekonstrukció

c értéket nem tudjuk (ha más c szerinti szeletet vizsgálnák, a post hoc szorzás miatt értelemszerűen gyönyörűen változna a kontraszt reprezentáció). Az ábrán tehát a $c = 1$ menti szelet látható.

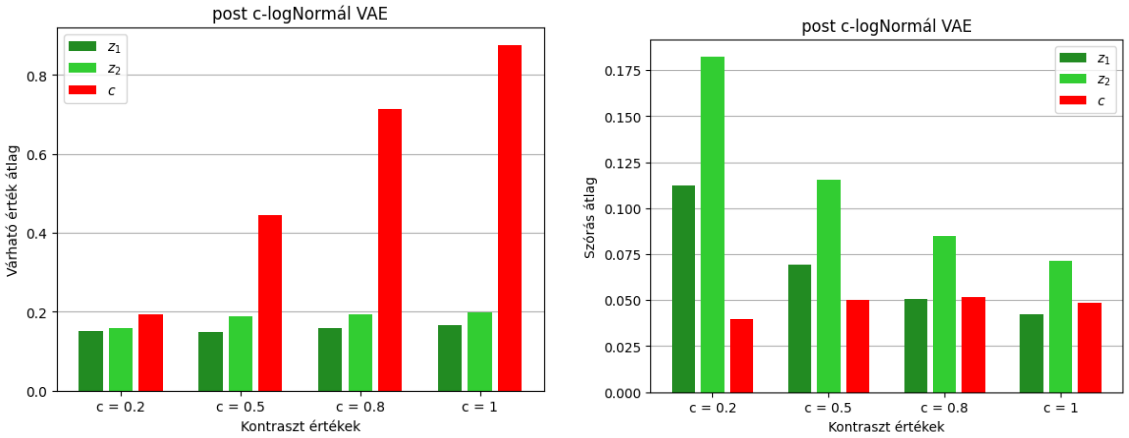
A számok kontrasztja elég hasonló egymáshoz, vagyis úgy látszik, hogy sikerült a kontrasztinvariáns elkódolás. A különböző számalakok folytonosan transzformálódnak egymásba, bár néhány helyen értelmezhetetlen értéket kapunk (például bal alsó sarok). Továbbá a számok alakja és határai elmosódottak, főleg a középső részeken. Az kontrasztra invariáns z reprezentáció örömteli, ezt kvantitatívan is ellenőrizzük.



4.15. ábra. Post hoc logNormál SMVAE látens tér

A (4.16) ábrán látszik, hogy a kontraszt függvényében nézett várható érték átlagok megerősítik a kvantitatív megfigyelést, miszerint a z dimenziók invariánsak a kontrasztra - ez a mostani modellben sikerült először. Továbbá egy másik tanulság is adódik, amit a látens teret figyelve nem tudtunk megállapítani, hogy c várható értéke együtt változik a kontraszttal, ráadásul c értéke is a valódi kontrasztértékhez közelít (ami nem meglepő, hiszen a (z_1, z_2, c) reprezentációban magas kontrasztú képeket láttunk, és ezeket skálázza c utólag). A nagyobb kontrasztérték esetén nagyobb különbség van c értéke és a kontraszt között, erre lehetséges fejlesztés a prior logNormál eloszlás paramétereinek változtatása lehetne, hogy pontosabban illeszkedjen a kontraszt eloszlására.

A szórással kapcsolatban is teljesül az előzetes várakozásunk. A z dimenziókra nézve a kontraszt növekedésével csökken a szórás. Emellett az látszik, hogy c szórása független a kontraszt értékétől. A c-nek megfelelő értékek a logNormál eloszlás várható értéke és szórása szerint lettek számolva.



(a) Várható érték (b) Szórás

4.16. ábra. Post hoc logNormál SMVAE posterior várható értékének és szórásának átlaga a kontraszt függvényében

A pre hoc szorzás Decoder utáni, post hoc szorzásra való lecserélése tehát javított a modell teljesítményén, és a legjobb reprezentációt eredményezte. A scale mixture lehetőségek közül a normál és gamma skálázást elvetettem, a lognormál eloszlás bizonyult a legjobbnak. A rekonstrukció nem volt mindig tökéletes, és a magas kontraszt értékek sem tudtak teljesen leképeződni. Ezek javítása érdekében több z látens dimenzió használatát, illetve a c prior paramétereinek finomhangolását lehet érdemes megpróbálni. Ezen túl viszont sikerült elérni a kitűzött célt, a kontrasztinvariáns reprezentációt. Az eredmények alapján azt a tanulságot vonhatjuk le, hogy ezt a modellt érdemes a természetes képek esetére továbbfejleszteni.

4.5. A modellek teljesítményének összehasonlítása

A korábbi fejezetekben négy különböző modell tulajdonságait vizsgáltuk: kvalitatívan a rekonstrukciót és a látens tér tulajdonságait, illetve kvantitatívan a posterior várható értékének és szórásának átlagát a kontraszt függvényében. Ez utóbbi elemzés segítségével megállapítottuk, hogy a post hoc logNormál SMVAE volt képes pontos kontrasztinvariáns reprezentációt létrehozni a látens tér z koordinátáira, illetve a kontraszt értéket elkódolni a c dimenzióban. A Standard VAE modellben láttunk még érdekes és pontos reprezentációt a kontrasztra, ahol egy radiális elrendezés-

ben a z vektor origótól való távolsága jelentette a kontrasztosság mértékét (itt nem volt megkülönböztetett látens c koordináta). A modellek teljesítményét az ELBO mérte a tanítás során, ami objektív összehasonlítás alapja is lehet. A (4.1) táblázat foglalja össze az ELBO változását a különböző modellek esetén. Az ELBO rekonstrukciós tagja miden modell esetében a BCE veszteségfüggvény volt, a regularizációs tag pedig minden esetben a modellnek megfelelő. A számolt értékek a teljes, 40000 nagyságú teszthalmazon egy képre vett átlagok.

	ELBO	Rekonstrukció	Regularizáció
Standard VAE	118.4	110.9	7.48
pre Normál SMVAE	127.1	119.3	7.73
pre Gamma(1) SMVAE	129.2	123.1	6.11
pre Gamma(3) SMVAE	129.3	123.3	6.03
post logNormál SMVAE	122.2	114.4	7.77

4.1. táblázat. A modellek ELBO-jának összehasonlítása

Az ELBO szerinti legjobb teljesítményt tehát a Standard VAE nyújtotta, ami megerősíti azt, hogy a polárkoordinátás reprezentációban is rejlenek lehetőségek. A logNormál SMVAE pedig ebben a mutatóban is megelőzi a többi, szétválasztott (*disentangled*) látens tér céljával alkotott modellt. A regularizációs tag elég kicsi a rekonstrukcióshoz képest, azonban a látens terek megfigyelésekor azt láttam, hogy nincs gond a reprezentáció regularitásával. Ellenkező esetben egy λ skálázó faktorral lehet nagyobb súlyt adni a regularizációs tagnak.

5. Összefoglalás

Célom a dolgozatban egy olyan neurális háló struktúra megalkotása volt, amely hatékonyan képes kontrasztinvariáns reprezentációt tanulni képi bemenetre. A képek statisztikájának és tulajdonságainak kifejezésére a Variational Autoencoder nevű modellt választottam. A VAE ugyanis explicit módon képes modellezni az adathalmaz ismeretlen eloszlását, és alkalmas arra, hogy a látens téren *disentangled*, azaz jelentés szerint szétválasztott reprezentációt tanuljon. A megvalósított modelleket a c-MNIST adathalmazon, a MNIST kontraszttal augmentált változatán tanítottam. A modellek megvalósításában inspirációm volt Wainwright és Simoncelli (1999) Gaussian Scale Mixtures nevű, eloszlásokat vegyítő eljárása - az itt szereplő szorzással valósítottam meg a kontraszt egyik látens koordinátára való leképezésére szolgáló induktív torzítást.

A munka során három eltérő felépítésű modellt alkottam meg. Elsőként a (4.2) részben egy standard VAE-t valósítottam meg, itt még nem volt cél, hogy a kont-

raszt egy látens dimenzióra legyen elkódolva. Ez a modell nyújtotta az ELBO szerinti legjobb teljesítményt, a kontrasztot pedig a látens z vektor hossza tárolta, ami egy izgalmas eredmény. Ezután azzal a céllal, a látens tér egy koordinátája reprezentálja a kontrasztot, a többi pedig a számalakot, először a *pre hoc* Scale Mixtures VAE-t (SMVAE) hoztam létre. Itt a látens koordináták közti szorzással szerettem volna jelentést adni a különböző dimenzióknak. A modell nem mutatott jó eredményeket, mivel a Decoder előtti szorzás implicit egy nagyítás szerinti összefüggést vitt a z és c látens részek közé. Ennek megoldására született meg a *post hoc* SMVAE. Itt a c dimenzióval való szorzás a Decoder kimenete utánra került, és a nemlinearitás miatt ez valódi függetlenséget eredményezett. Ez a modell nyújtotta az ELBO szerinti második legjobb teljesítményt, és jól megvalósította a számalak és kontraszt szétválasztott reprezentációját.

A c-MNIST adathalmazon a modellek egy továbbfejlesztési lehetősége a Loaiza-Ganem és Cunningham (2019) által leírt, folytonos pixelértékekre optimalizált BCE veszteségfüggvény használata, az elmosódott rekonstrukció javítása érdekében. Természetes képek esetében valószínűsíthetőleg jobb teljesítményt nyújt az MSE, ezért erre a továbbfejlesztésre ott minden bizonnyal nem lesz szükség. A Standard VAE látens terében megjelenő radiális reprezentáció alkalmas lehet további kutatásra. A c-MNIST esetében diszkrét kontraszttal dolgoztam, a modellek reprezentációja ezt folytonossá tette, azonban a valósághoz jobban passzolhat, ha egy uniform vagy más eloszlású, folytonos valószínűségi változóval szorozzuk az eredeti adathalmazt. Végül (a természetes képek esetében is) további két változtatást lehet érdemes megfontolni. Az egyik a Rezende és Mohamed (2015) által leírt *normalizing flow* használata, ami pontosabb posterior eloszlás megalkotásával javítja a modell eredményeit. A másik pedig a látens tér szeparáltságának a bemutatott hisztogramokon túli mérőszáma, a Whittington, Dorrell, Ganguli, és Behrens (2023) által használt eljárás szerint.

A kutatás célja az volt, hogy az egyszerűbb c-MNIST adathalmazon egy olyan modellt alkosson meg, ami alkalmas lehet természetes képek tanulására való továbbfejlesztésre. Ez jóval bonyolultabb adathalmaz, és így a reprezentációban a képek pixelszámával megegyező dimenziójú látens térre van szükség a hatékony tanuláshoz. Egy ilyen modell tanítása idő- és erőforrásigényes, ezért volt fontos, hogy előtte egy egyszerűbb adathalmazon megérthessem a VAE működését, és megfigyeljem, hogy milyen modell képes hatékonyan kontrasztinvariáns reprezentációt tanulni. A továbbfejlesztés irányában a legígéretesebb modellnek a *post hoc* SMVAE bizonyult, c-re nézve logNormál priorral. Az utólagos szorzással megvalósult a látens változók kontraszt szerinti függetlensége, és a modell rekonstrukciós szempontból is megfelelően teljesített, így az eredményeim szerint ezt érdemes a természetes képek elemzésére felhasználni.

Hivatkozások

Bishop, C. M. (1994). Neural networks and their applications. *Review of Scientific Instruments*, *65*(6), 1803–1832.

Buccigrossi, R. W., & Simoncelli, E. P. (1999). Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing*, *8*(12), 1688–1701.

Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan*, *21*, 22.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., ... Saurous, R. A. (2017). TensorFlow Distributions. *CoRR*, *abs/1711.10604*. Retrieved from `http://arxiv.org/abs/1711.10604`

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). *Generative Adversarial Networks.* Retrieved from `https://arxiv.org/abs/1406.2661`

Graving, J. M., & Couzin, I. D. (2020). VAE-SNE: a Deep Generative model for simultaneous dimensionality reduction and clustering. *bioRxiv*. Retrieved from `https://www.biorxiv.org/content/early/2020/07/17/2020.07.17.207993`

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... others (2018). Recent advances in Convolutional Neural Networks. *Pattern recognition*, *77*, 354–377.

Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. *arXiv*. Retrieved from `https://arxiv.org/abs/1312.6114`

Kingma, D. P., & Welling, M. (2019). An introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with Deep Convolutional Neural Networks. *Communications of the ACM*, *60*(6), 84–90.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

LeCun, Y., Cortes, C., & Burges, C. (n.d.). MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*.

Loaiza-Ganem, G., & Cunningham, J. P. (2019). The continuous Bernoulli: fixing a pervasive error in Variational Autoencoders. *Advances in Neural Information Processing Systems*, *32*.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607–609.

PyTorch. (2023). *Probability distributions - torch.distributions - pytorch 2.0 documentation.* Retrieved from `https://pytorch.org/docs/`

stable/distributions.html?highlight=distribution#module-torch.distributions

Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. *Proceedings of Machine Learning Research*, *37*, 1530–1538.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv*. Retrieved from http://arxiv.org/abs/1609.04747

Wainwright, M. J., & Simoncelli, E. (1999). Scale mixtures of Gaussians and the statistics of natural images. *Advances in Neural Information Processing Systems*, *12*, 855–861.

Whittington, J. C. R., Dorrell, W., Ganguli, S., & Behrens, T. E. J. (2023). *Disentanglement with biological constraints: A theory of functional cell types.* Retrieved from https://arxiv.org/abs/2210.01768