

compressie-algoritme ontwerpen?

compressie-algoritme ontwerpen?



Auto-encoders voor compressie

Josse Van Delm

December 13, 2018

Overzicht

Overzicht

■ Introductie

Overzicht

- Introductie
- Auto-encoder

Overzicht

- Introductie
- Auto-encoder
- Training

Overzicht

- Introductie
- Auto-encoder
- Training
- Resultaten

Als het bij beeldverwerking lukt,....

Als het bij beeldverwerking lukt,....

- klassieke beeldverwerking

Als het bij beeldverwerking lukt,....

- klassieke beeldverwerking
- machine learning - shallow learning

Als het bij beeldverwerking lukt,....

- klassieke beeldverwerking
- machine learning - shallow learning
- neural networks - deep learning

Als het bij beeldverwerking lukt,....

- klassieke beeldverwerking
- machine learning - shallow learning
- neural networks - deep learning

...werkt het dan ook bij compressie?

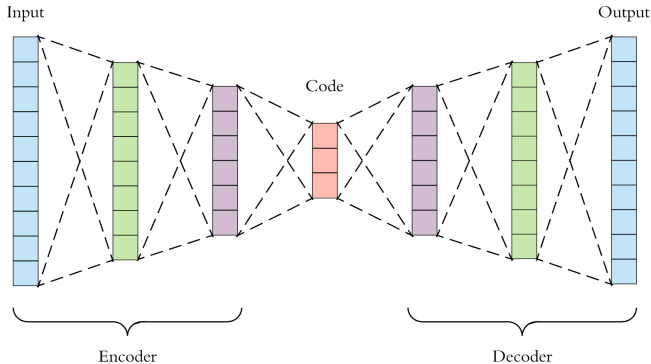
Als het bij beeldverwerking lukt,....

- klassieke beeldverwerking
- machine learning - shallow learning
- neural networks - deep learning

...werkt het dan ook bij compressie?

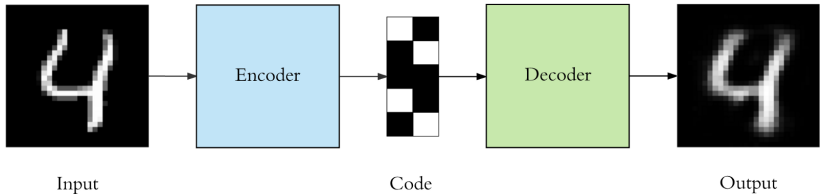
Kunnen we iets maken dat beter presteert dan jpeg?

Een auto-encoder doet aan unsupervised learning



Bron: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>

We gebruiken de latent space als compressedata



Bron: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>

We gebruiken de latent space als compressedata

Het idee:

We gebruiken de latent space als compressedata

Het idee:

- Train auto-encoder

We gebruiken de latent space als compressedata

Het idee:

- Train auto-encoder
- Splits auto-encoder op

We gebruiken de latent space als compressedata

Het idee:

- Train auto-encoder
- Splits auto-encoder op
- Encoder: afbeelding \rightarrow latent space

We gebruiken de latent space als compressedata

Het idee:

- Train auto-encoder
- Splits auto-encoder op
- Encoder: afbeelding \rightarrow latent space
- Decoder: latent space \rightarrow afbeelding

De cpu is te traag, en installatie is vervelend

We gebruikten:

De cpu is te traag, en installatie is vervelend

We gebruikten:

- Keras deep learning framework frontend

De cpu is te traag, en installatie is vervelend

We gebruikten:

- Keras deep learning framework frontend
- Tensorflow backend

De cpu is te traag, en installatie is vervelend

We gebruikten:

- Keras deep learning framework frontend
- Tensorflow backend
- Jupyter Notebook (interactive python shell)

De cpu is te traag, en installatie is vervelend

We gebruikten:

- Keras deep learning framework frontend
- Tensorflow backend
- Jupyter Notebook (interactive python shell)
- Matplotlib (MATLAB-style plotting)

De cpu is te traag, en installatie is vervelend

We gebruikten:

- Keras deep learning framework frontend
- Tensorflow backend
- Jupyter Notebook (interactive python shell)
- Matplotlib (MATLAB-style plotting)
- Docker Container

De cpu is te traag, en installatie is vervelend

We gebruikten:

- Keras deep learning framework frontend
- Tensorflow backend
- Jupyter Notebook (interactive python shell)
- Matplotlib (MATLAB-style plotting)
- Docker Container

implementatie op github!

Ons netwerk is erg simpel

Het bestaat uit:

Ons netwerk is erg simpel

Het bestaat uit:

- *Input* : afbeelding = $28 \cdot 28 \cdot 1 = 784$ waarden (grijswaarden)

Ons netwerk is erg simpel

Het bestaat uit:

- *Input* : afbeelding = $28 \cdot 28 \cdot 1 = 784$ waarden (grijswaarden)
- *Encoder* : $3 \cdot (\text{Convolution} + \text{Max pooling})$

Ons netwerk is erg simpel

Het bestaat uit:

- *Input* : afbeelding = $28 \cdot 28 \cdot 1 = 784$ waarden (grijswaarden)
- *Encoder* : $3 \cdot (\text{Convolution} + \text{Max pooling})$
- *Latent space*: tensor = $4 \cdot 4 \cdot 8 = 128$ waarden

Ons netwerk is erg simpel

Het bestaat uit:

- *Input* : afbeelding = $28 \cdot 28 \cdot 1 = 784$ waarden (grijswaarden)
- *Encoder* : $3 \cdot (\text{Convolution} + \text{Max pooling})$
- *Latent space*: tensor = $4 \cdot 4 \cdot 8 = 128$ waarden
- *Decoder* : $3 \cdot (\text{Convolution} + \text{Upsampling})$

Ons netwerk is erg simpel

Het bestaat uit:

- *Input* : afbeelding = $28 \cdot 28 \cdot 1 = 784$ waarden (grijswaarden)
- *Encoder* : 3 · (Convolution + Max pooling)
- *Latent space*: tensor = $4 \cdot 4 \cdot 8 = 128$ waarden
- *Decoder* : 3 · (Convolution + Upsampling)
- *Output* : afbeelding = $28 \cdot 28 \cdot 1 = 784$ waarden (grijswaarden)

Ons netwerk is erg simpel

Het bestaat uit:

- *Input* : afbeelding = $28 \cdot 28 \cdot 1 = 784$ waarden (grijswaarden)
- *Encoder* : $3 \cdot (\text{Convolution} + \text{Max pooling})$
- *Latent space*: tensor = $4 \cdot 4 \cdot 8 = 128$ waarden
- *Decoder* : $3 \cdot (\text{Convolution} + \text{Upsampling})$
- *Output* : afbeelding = $28 \cdot 28 \cdot 1 = 784$ waarden (grijswaarden)

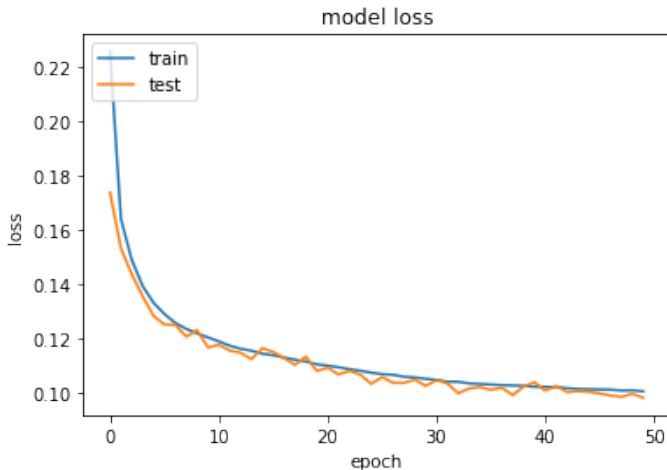
*We mikken dus op een compressieratio van: **6,125:1***

Ons doel is het verkleinen van de loss

We werken met de MNIST dataset



We kunnen zien dat de loss kleiner wordt



De cijfers lijken op elkaar → lage loss



De resultaten vallen een beetje tegen

Deze aanpak heeft veel nadelen:

De resultaten vallen een beetje tegen

Deze aanpak heeft veel nadelen:

- resultaten erg afhankelijk van trainingsdata

De resultaten vallen een beetje tegen

Deze aanpak heeft veel nadelen:

- resultaten erg afhankelijk van trainingsdata
- grote netwerken → veel computer-resources bij inferentie

De resultaten vallen een beetje tegen

Deze aanpak heeft veel nadelen:

- resultaten erg afhankelijk van trainingsdata
- grote netwerken → veel computer-resources bij inferentie
- Onze aanpak is erg beperkt ($28 \cdot 28$ grijswaarden)

De resultaten vallen een beetje tegen

Deze aanpak heeft veel nadelen:

- resultaten erg afhankelijk van trainingsdata
- grote netwerken → veel computer-resources bij inferentie
- Onze aanpak is erg beperkt ($28 \cdot 28$ grijswaarden)
- JPEG haalt gemakkelijk grotere compressieratios!

De resultaten vallen een beetje tegen

Deze aanpak heeft veel nadelen:

- resultaten erg afhankelijk van trainingsdata
- grote netwerken → veel computer-resources bij inferentie
- Onze aanpak is erg beperkt ($28 \cdot 28$ grijswaarden)
- JPEG haalt gemakkelijk grotere compressieratios!

Demo!

Wel,... ongeveer :(



Bedankt voor jullie aandacht! Vragen?