Deep Learning

Knowledge Discovery and Data Mining 2 (VU) (706.715)

Kevin Winter

Know Center GmbH

23-05-2019

Outline

- Introduction
- Neural Network architectures
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Autoencoder
  - Attention Mechanism
  - Generative Adverserial Networks
- Transfer Learning
- Interpretability

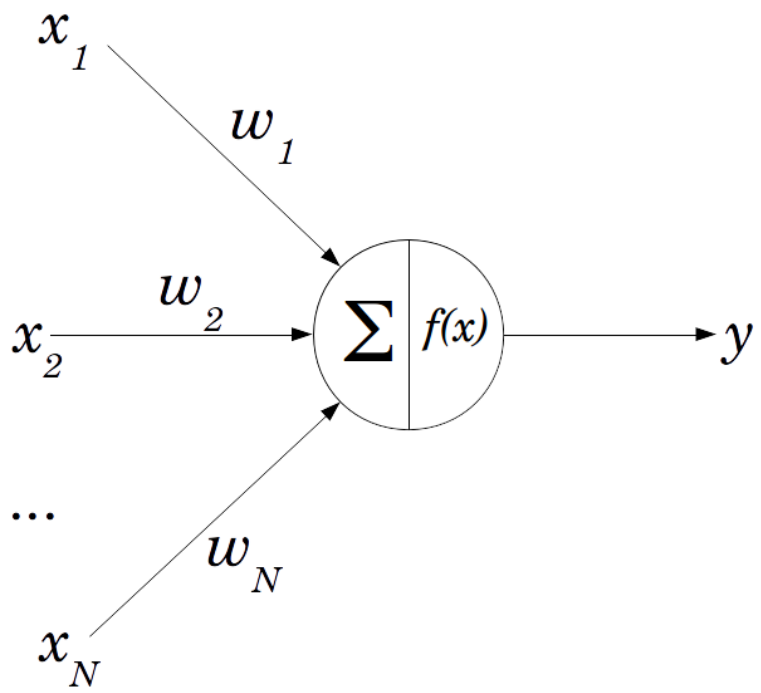
Prerequisites

- Neural Networks
  - Backpropagation

- Optimization
  - Basic algorithms e. g. Gradient Descent
  - Regularization (L1, L2 norm, Dropout, Early Stopping)

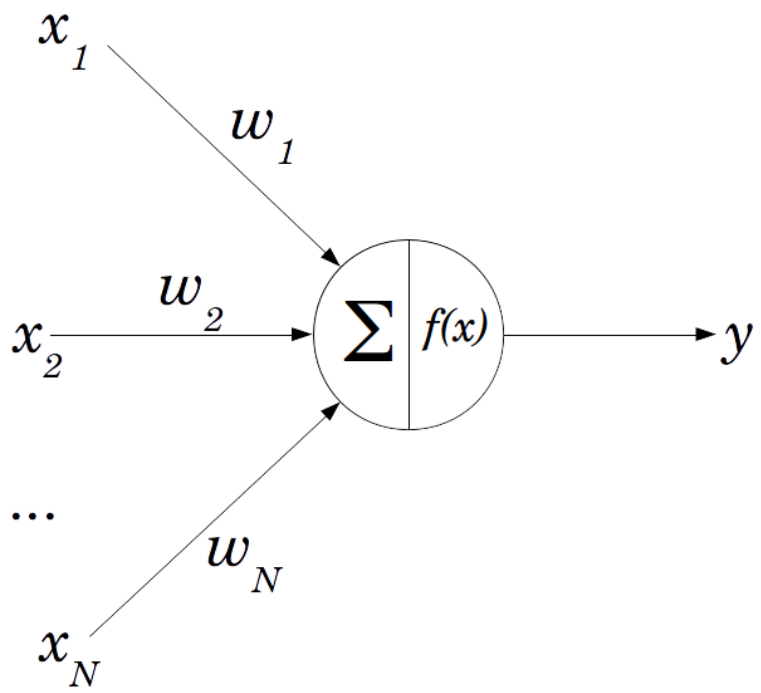
What is deep?

- First there was the perceptron



What is deep?

- First there was the perceptron



- If f(x) = x
  -> Linear Regression
- 1 hidden layer
  -> can approximate any function
- Why would we want more?

What is deep?

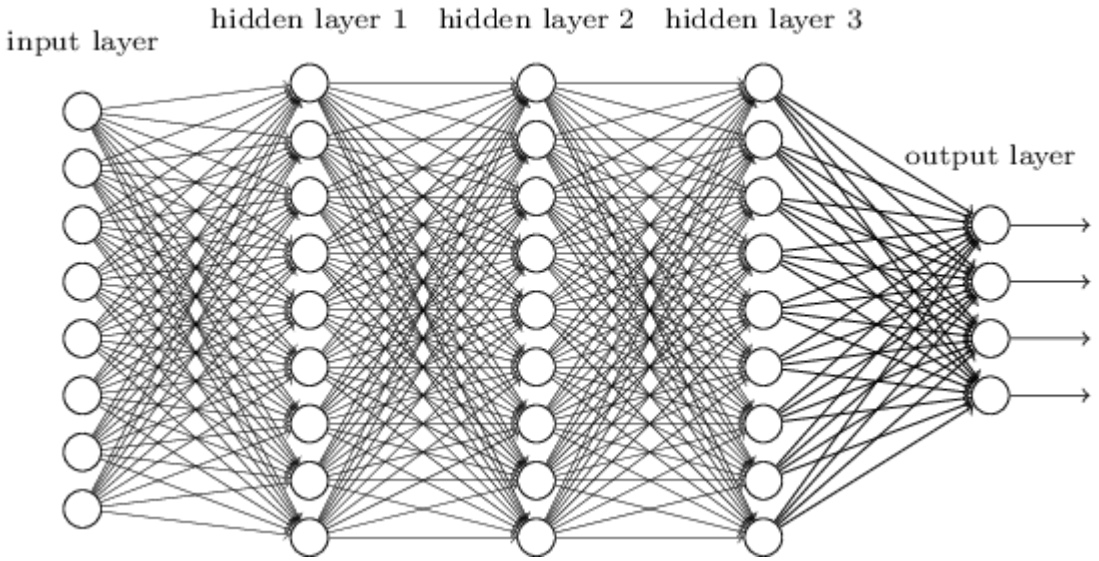
- Fully Connected Networks (FCN)

  Advantages:

  - Represent Feature Hierarchy
  - Can be faster to train
  - Potentially less connections

  Disadvantages:

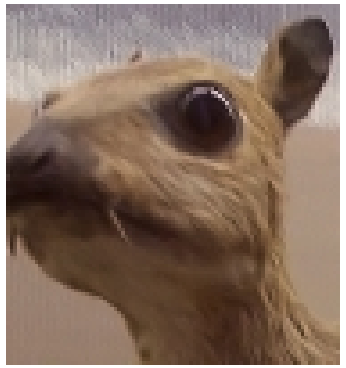
  - No prior knowledge induced
  - Still many connections



Convolutional Neural Network (CNN)

- Convolution Operation
- Applies same kernel (filter) over whole input space
- Computed efficiently on GPU

Input image



Convolution Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map

https://developer.nvidia.com/discover/convolution

Convolutional Neural Network (CNN)

- CNNs:
  - Idea: Use spatial dependencies to reduce connectivity and learn the kernels from data
  - Advantages:
    - Sparse connectivity
      Neurons receive input only from a local receptive field (RF)
    - Shared weights
      Each neuron computes the same function for each RF
    - Pooling
      Predefined function instead of learnt weights for some layers

Convolutional Neural Network (CNN)

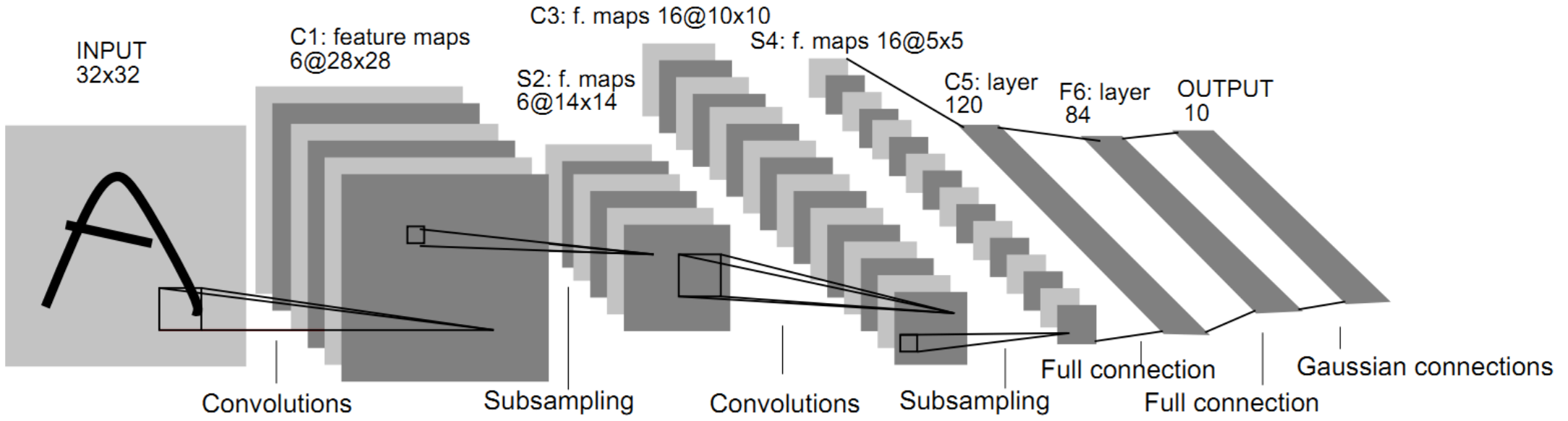
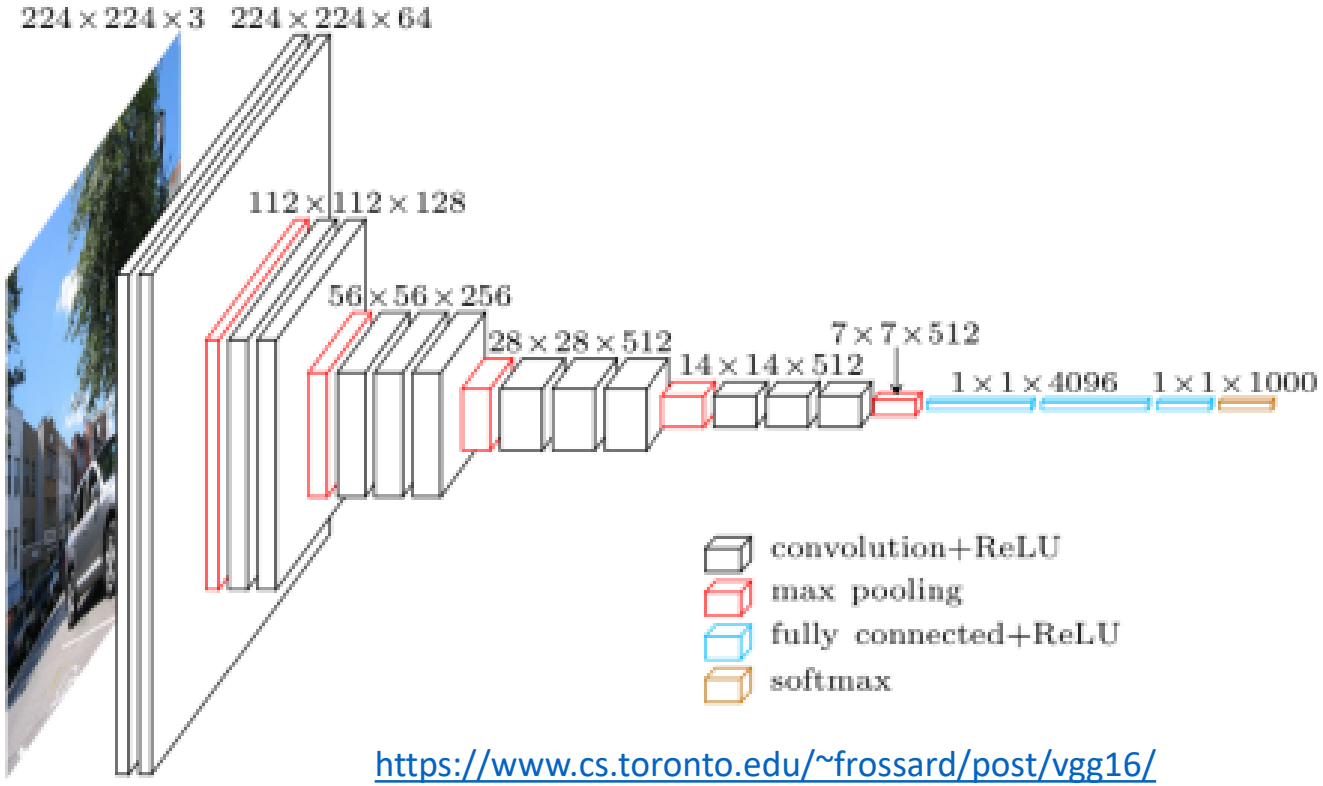


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

[16]

Convolutional Neural Network (CNN)

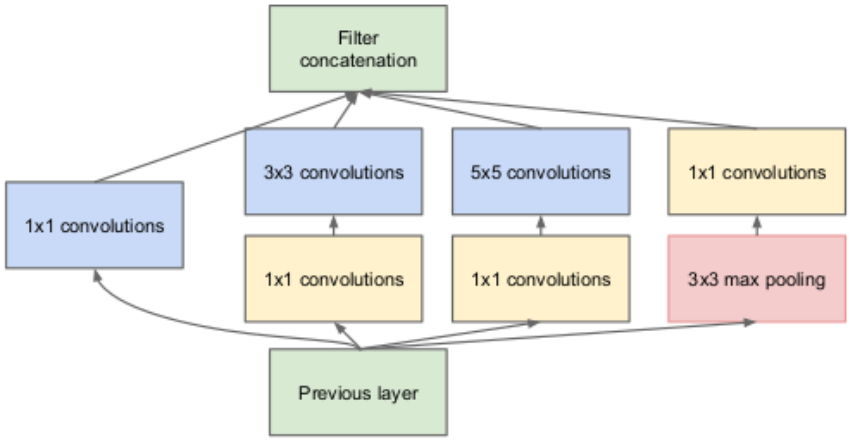
- VGG 16

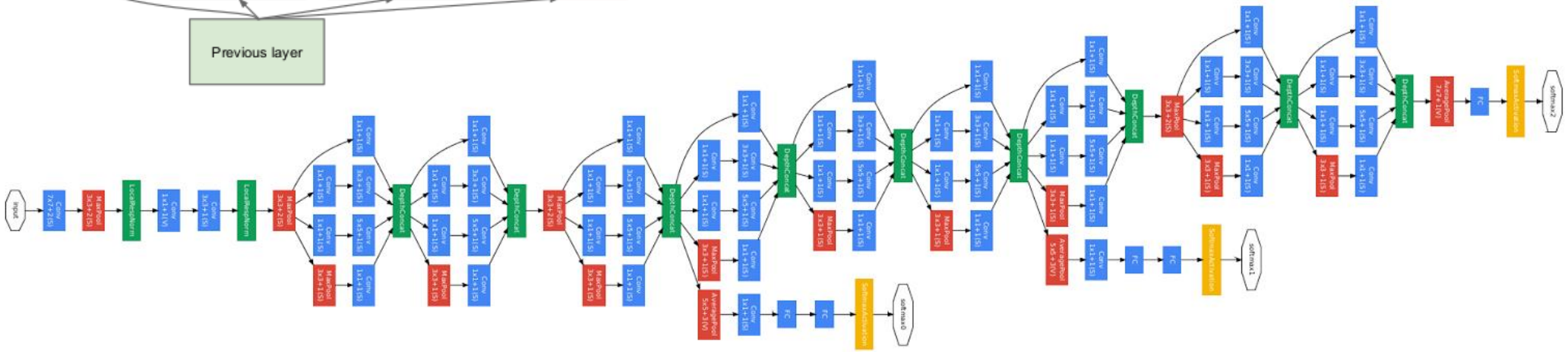


https://www.cs.toronto.edu/~frossard/post/vgg16/

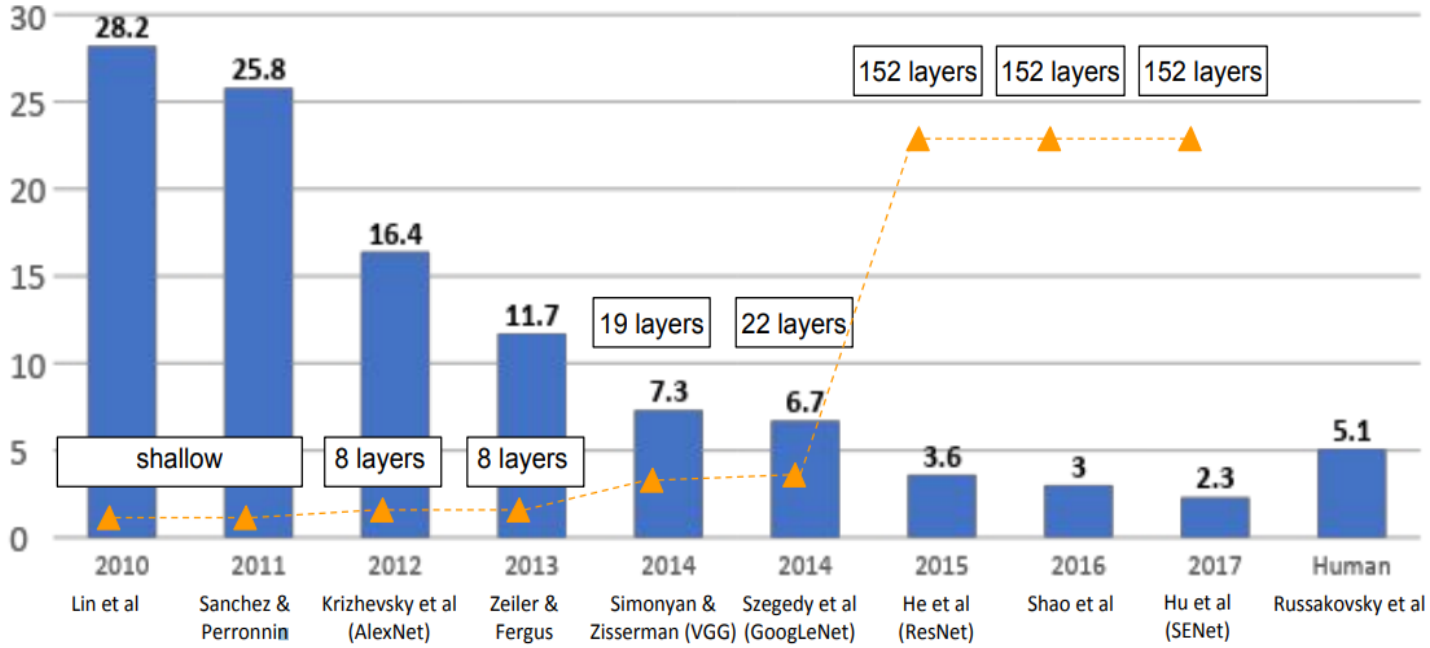
Convolutional Neural Network (CNN)

- Google's LeNet

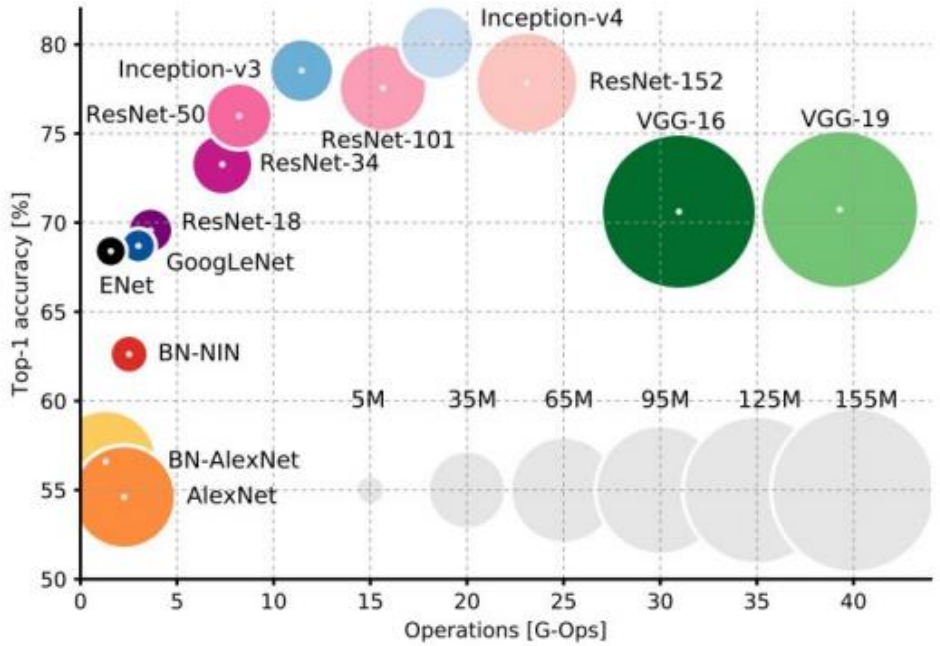




Convolutional Neural Network (CNN)



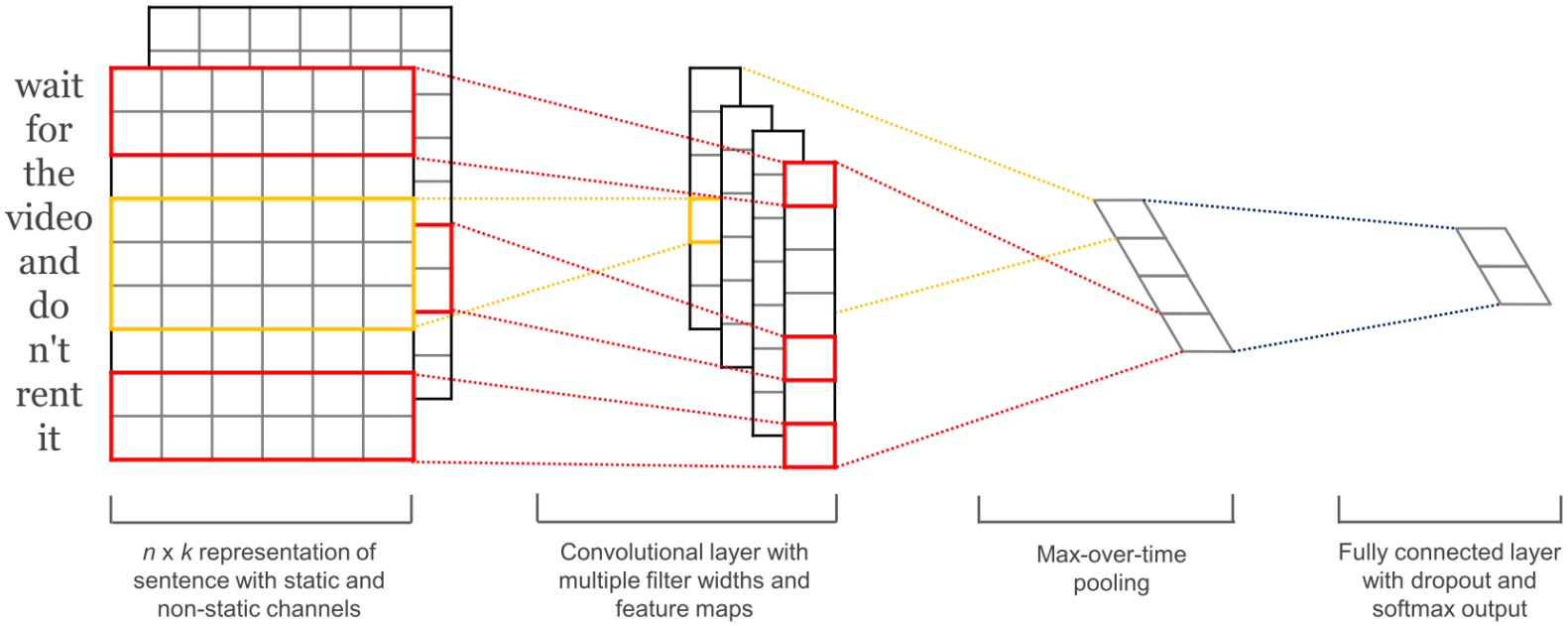
[1]



[2]

Convolutional Neural Network (CNN)

- Convolutions on text data

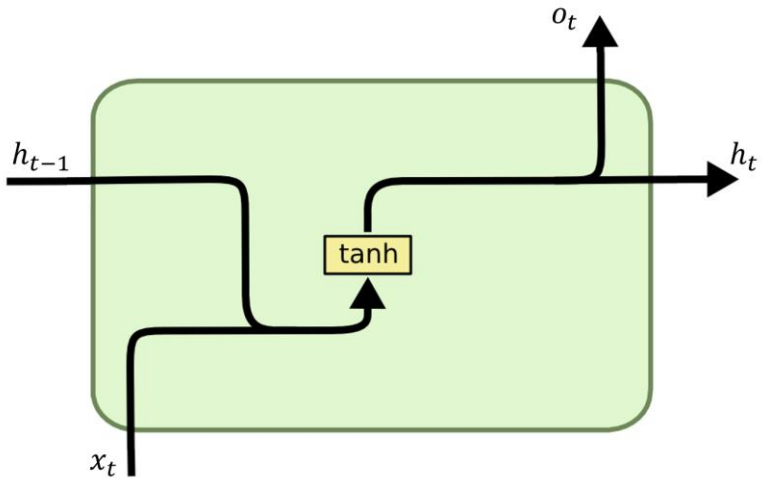


Recurrent Neural Network (RNN)

- Infinite depth!!

$$h_t = \sigma_h(i_t) = \sigma_h(U_h x_t + V_h h_{t-1} + b_h)$$

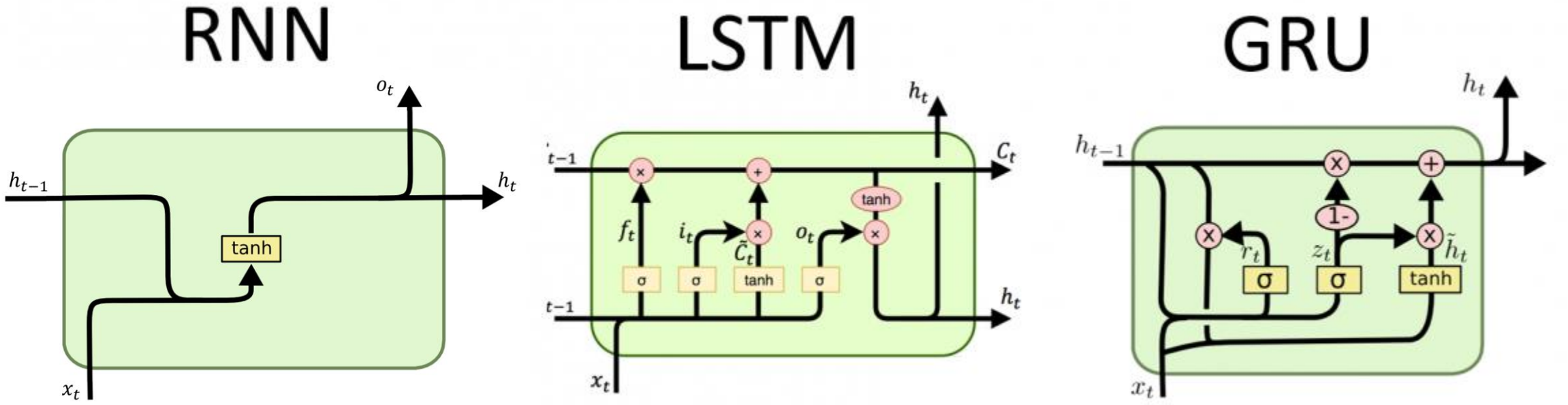
$$y_t = \sigma_y(a_t) = \sigma_y(W_y h_t + b_h)$$



[1]

Recurrent Neural Network (RNN)

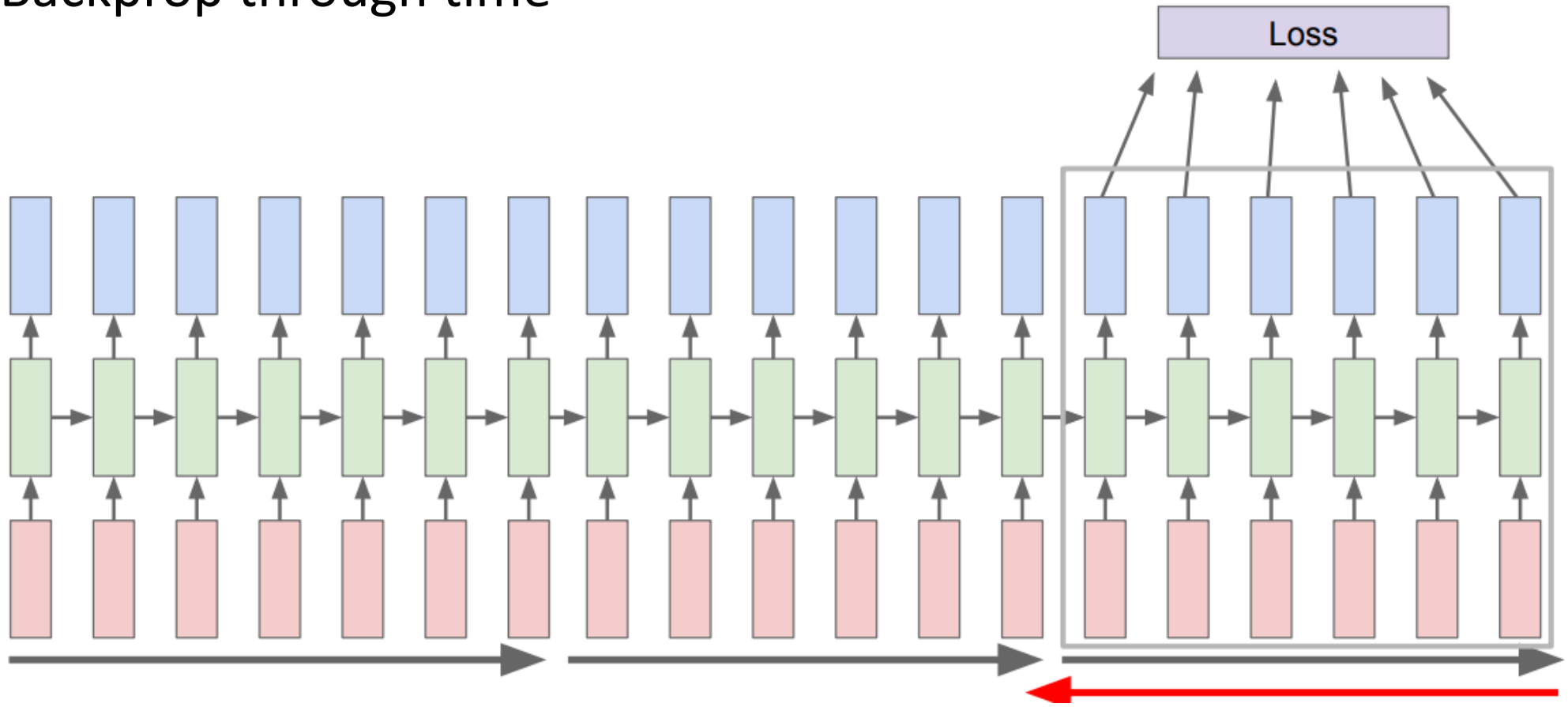
- Different cell types
  - Vanilla RNN
  - Long Short-Term Memory (LSTM)
  - Gated Recurrent Unit (GRU)



[1]

Recurrent Neural Network (RNN)

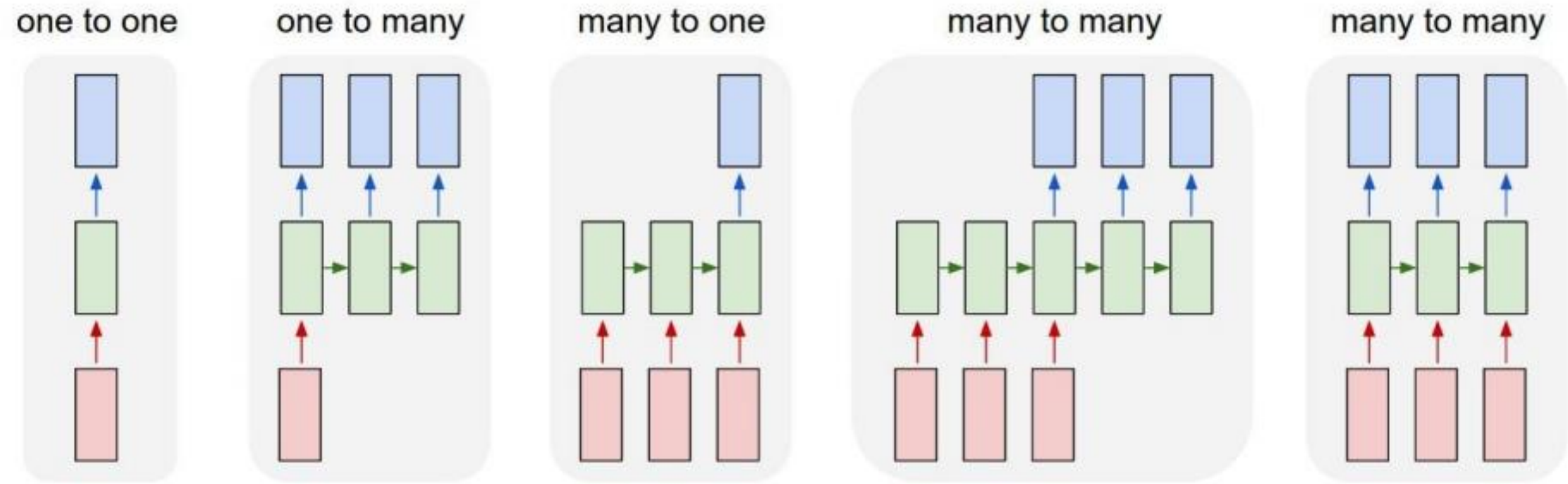
- Backprop through time



[1]

Recurrent Neural Network (RNN)

- Inputs and Outputs

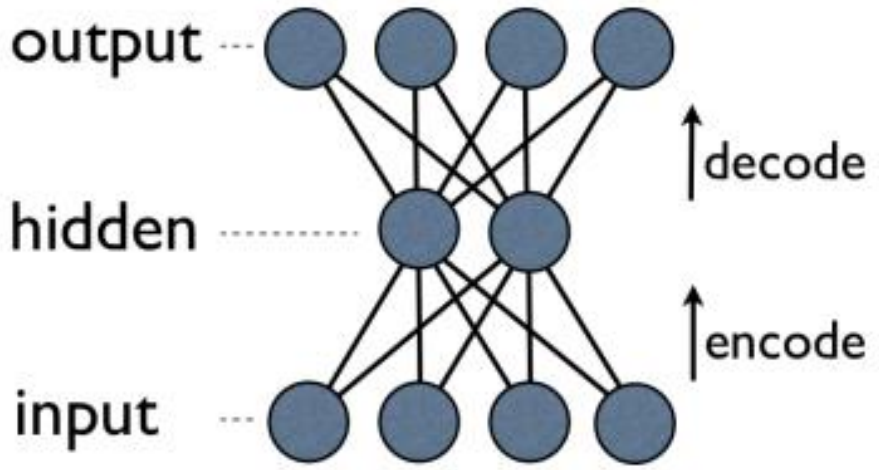


[1]

Autoencoder

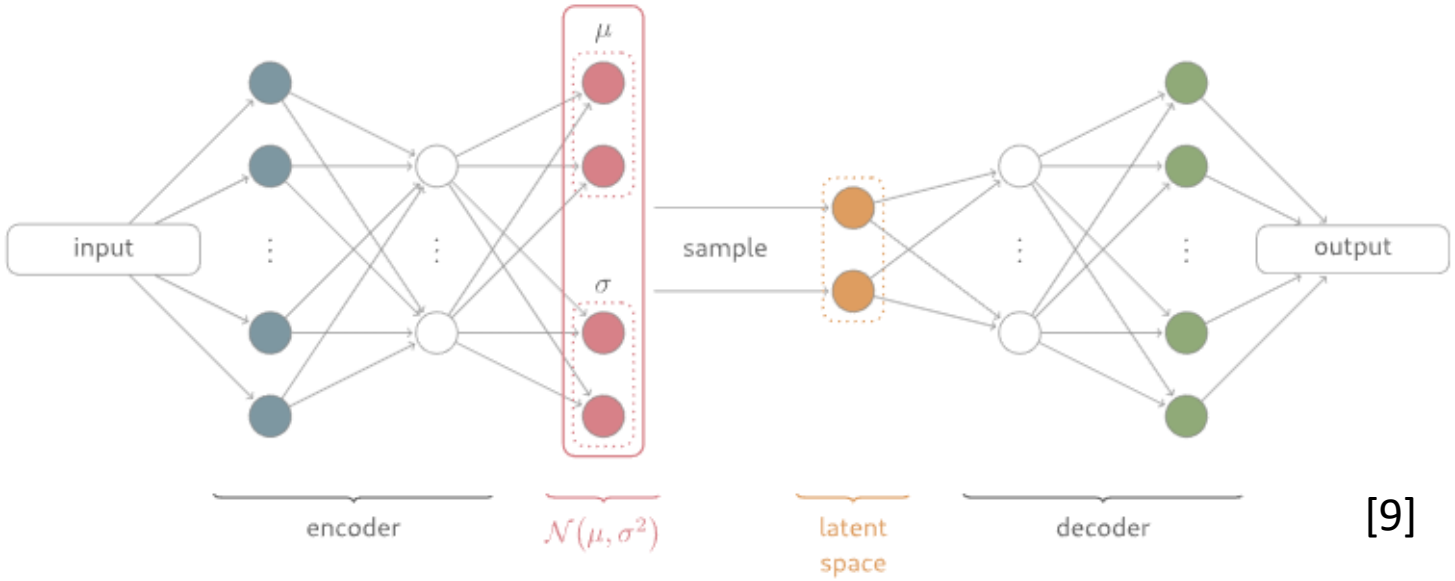
- Consists of two parts:
  - Encoder
  - Decoder

- Tries to replicate input
  - Unsupervised
  - Minimizes reconstruction error $\|x - \hat{x}\|^2$
  - Has to learn internal representation of the input



Autoencoder

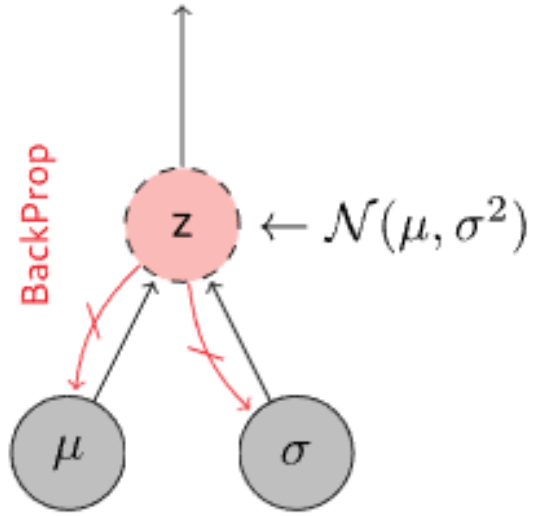
- Variational Autoencoder
  - Enforces normally distributed latent space
  - Maximize $l_i(\theta, \phi) = -\mathbb{E}_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i \mid z)] + \mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z))$



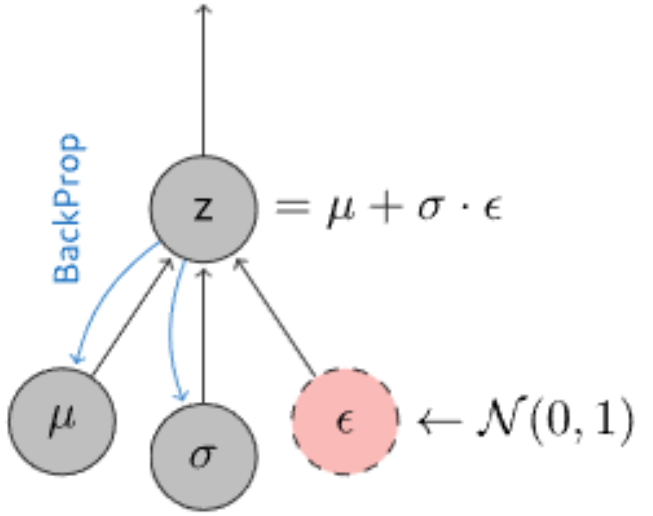
[9]

Autoencoder

- Variational Autoencoder
  - Reparameterization trick



(a) *Before applying reparameterization trick*

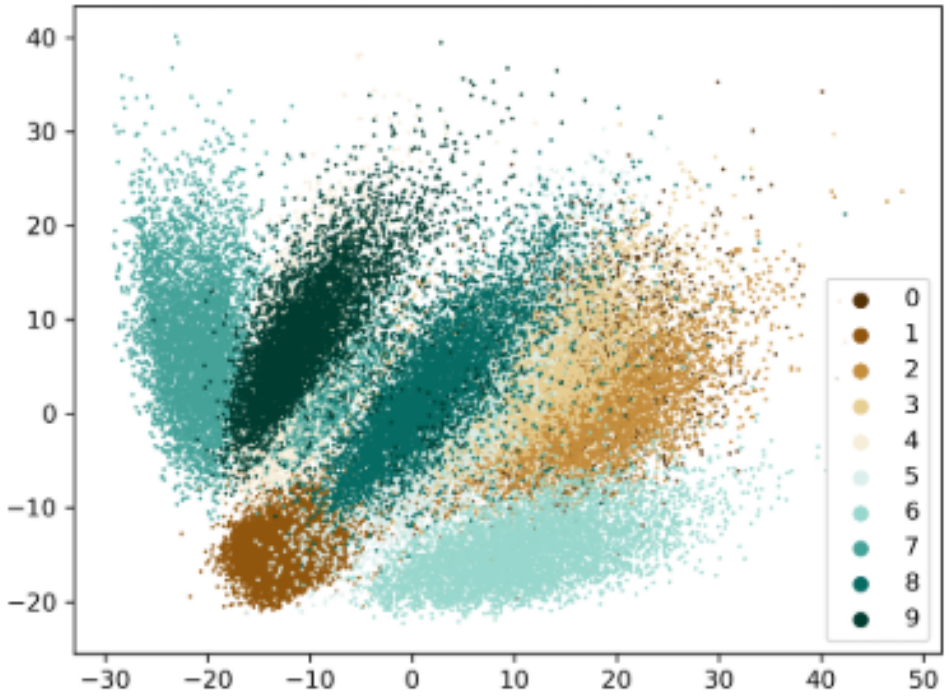


(b) *After applying reparameterization trick*

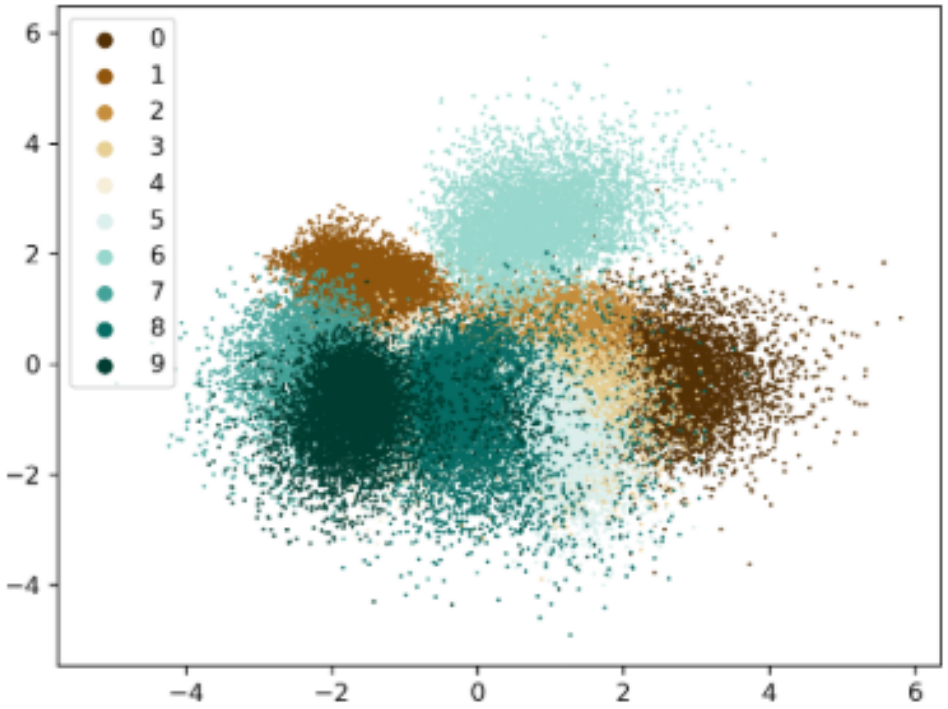
[9]

Autoencoder

- Variational Autoencoder
  - Latent Space



(a) *Latent Distribution by Label for AE*



(b) *Latent Distribution by Label for VAE* [9]

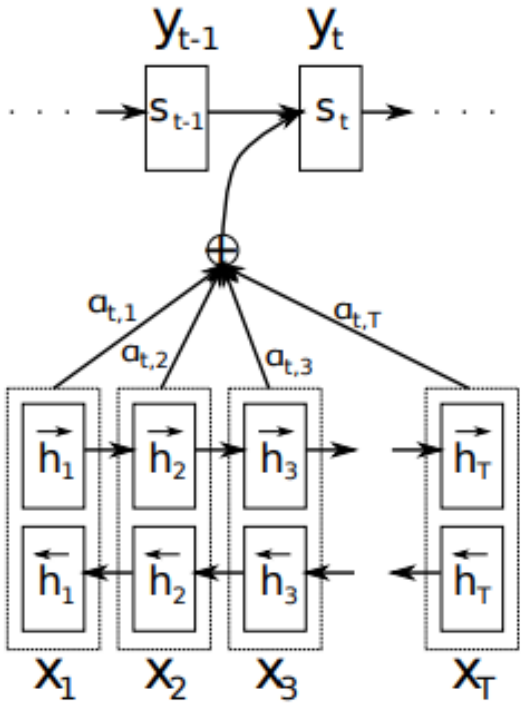
Attention

- Idea: Not all features are equally important for the task at hand
- Reweight features based on current focus

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

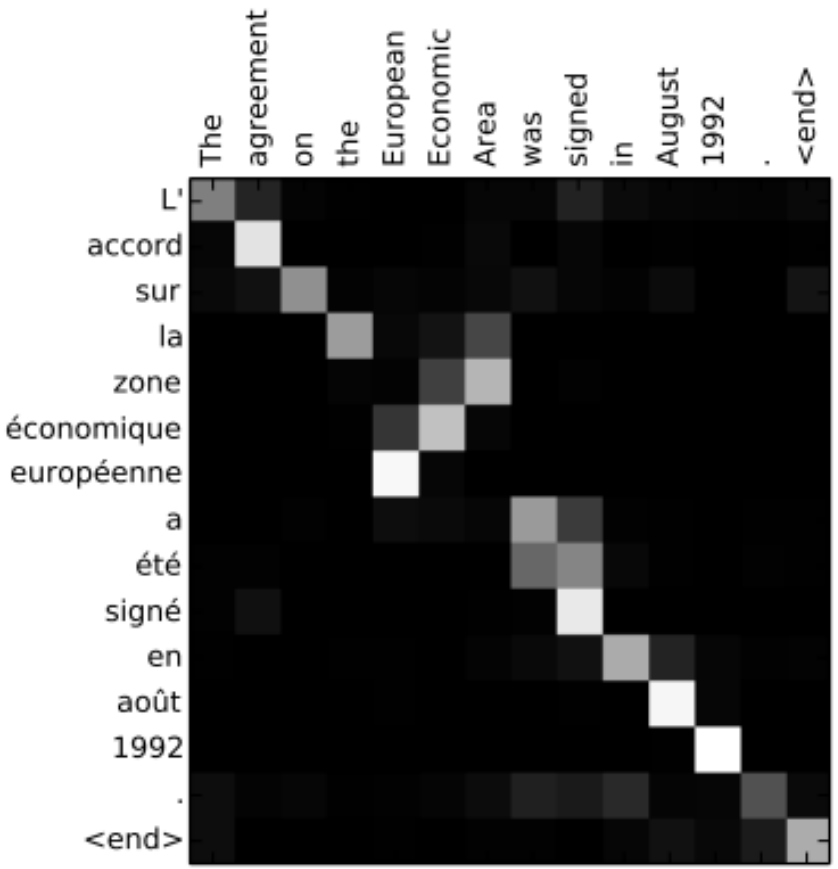
$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)},$$

$$e_{ij} = a(s_{i-1}, h_j)$$



[6]

Attention

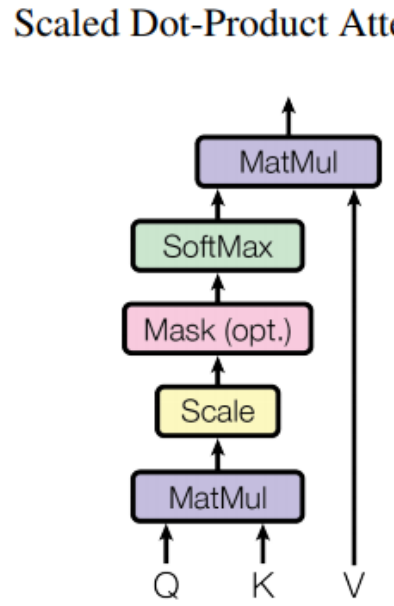


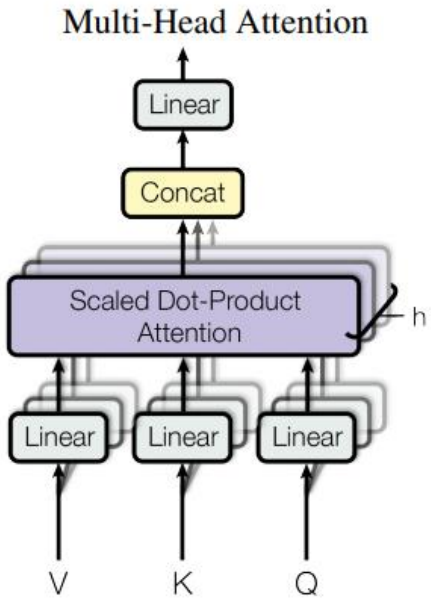
[6]

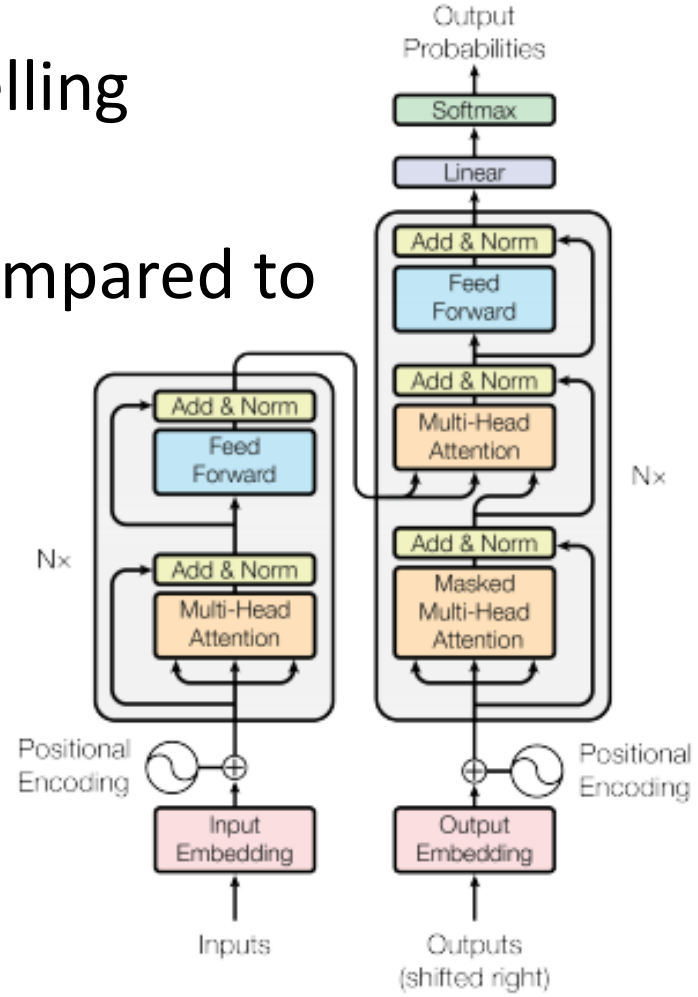
[7]

Transformer Network

- Use Attention only for sequence to sequence modelling (translation)
- Considerably reduced computational complexity compared to RNNs



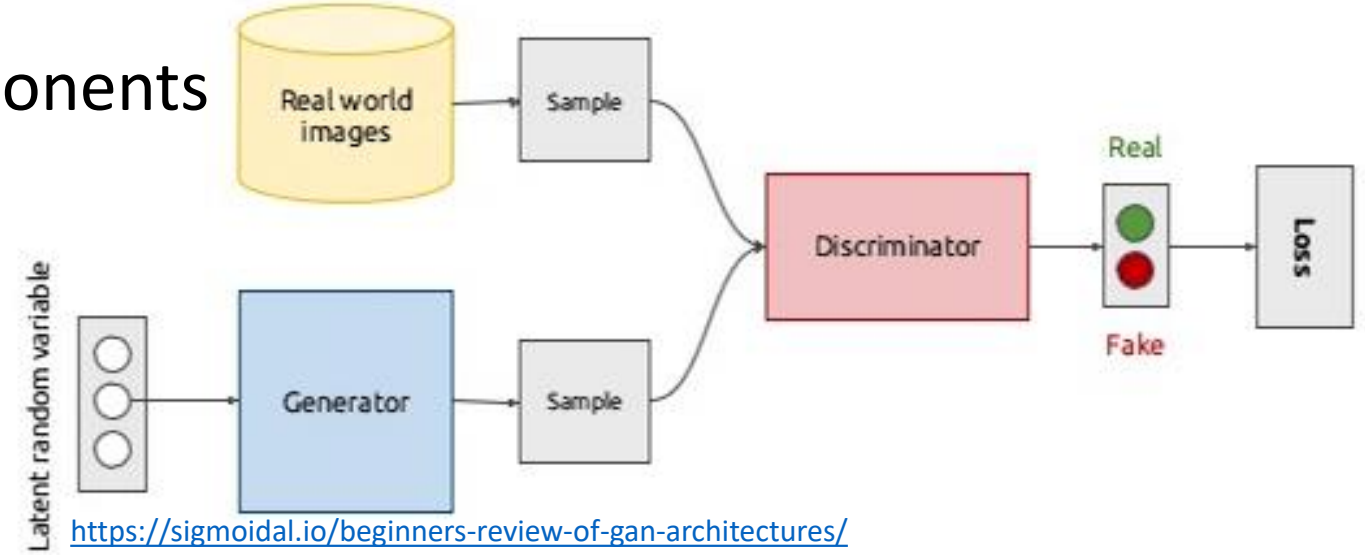




[8]

Generative Adversarial Network (GAN)

- Consists of two main components
  - Generator G
  - Discriminator D



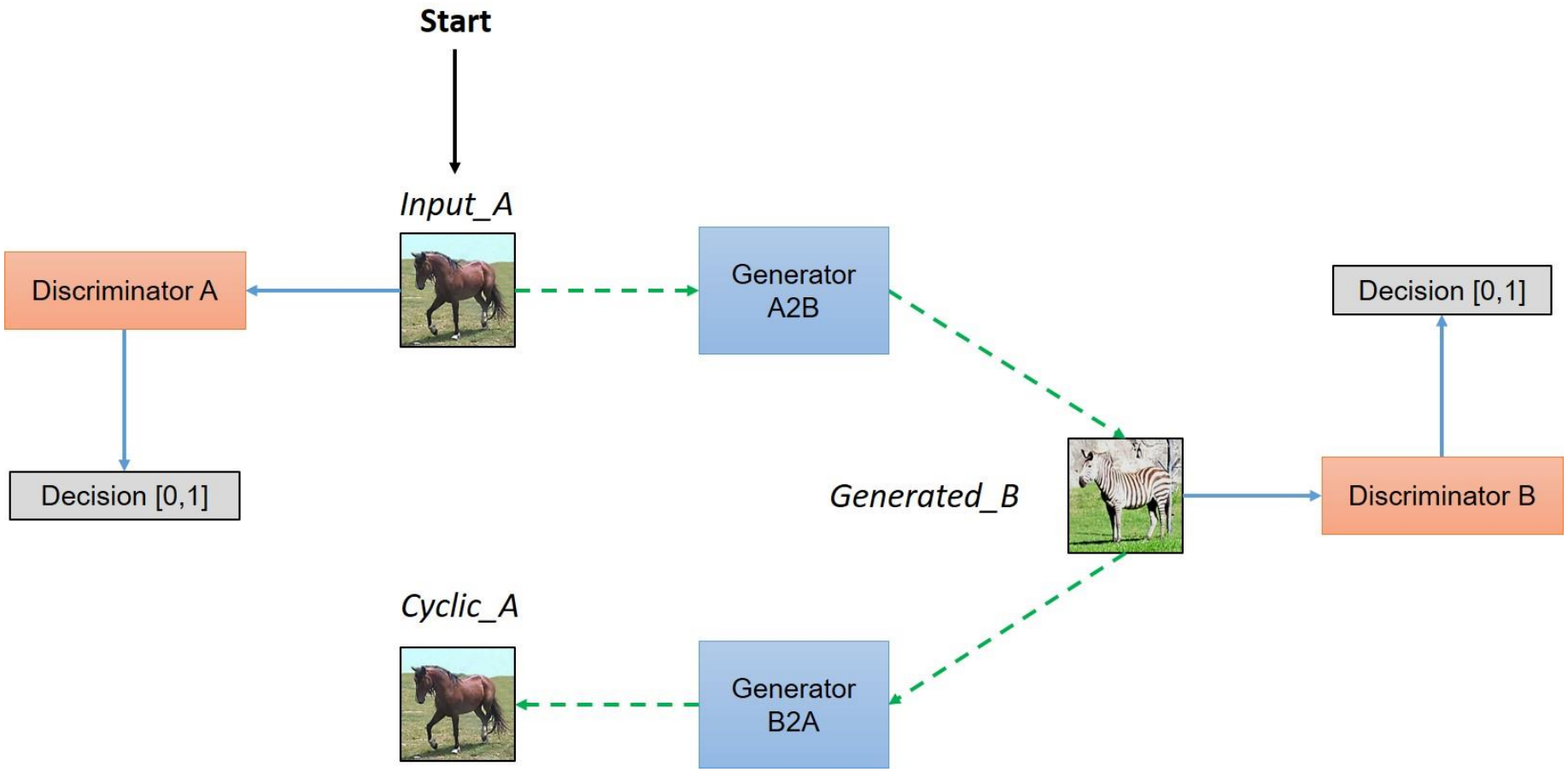
https://sigmoidal.io/beginners-review-of-gan-architectures/

- Generator generates samples from random noise
- Discriminator tries to distinguish between real and generated samples
- Opposing objectives -> Minimax game

$$J^{(D)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(z)\right)\right)$$

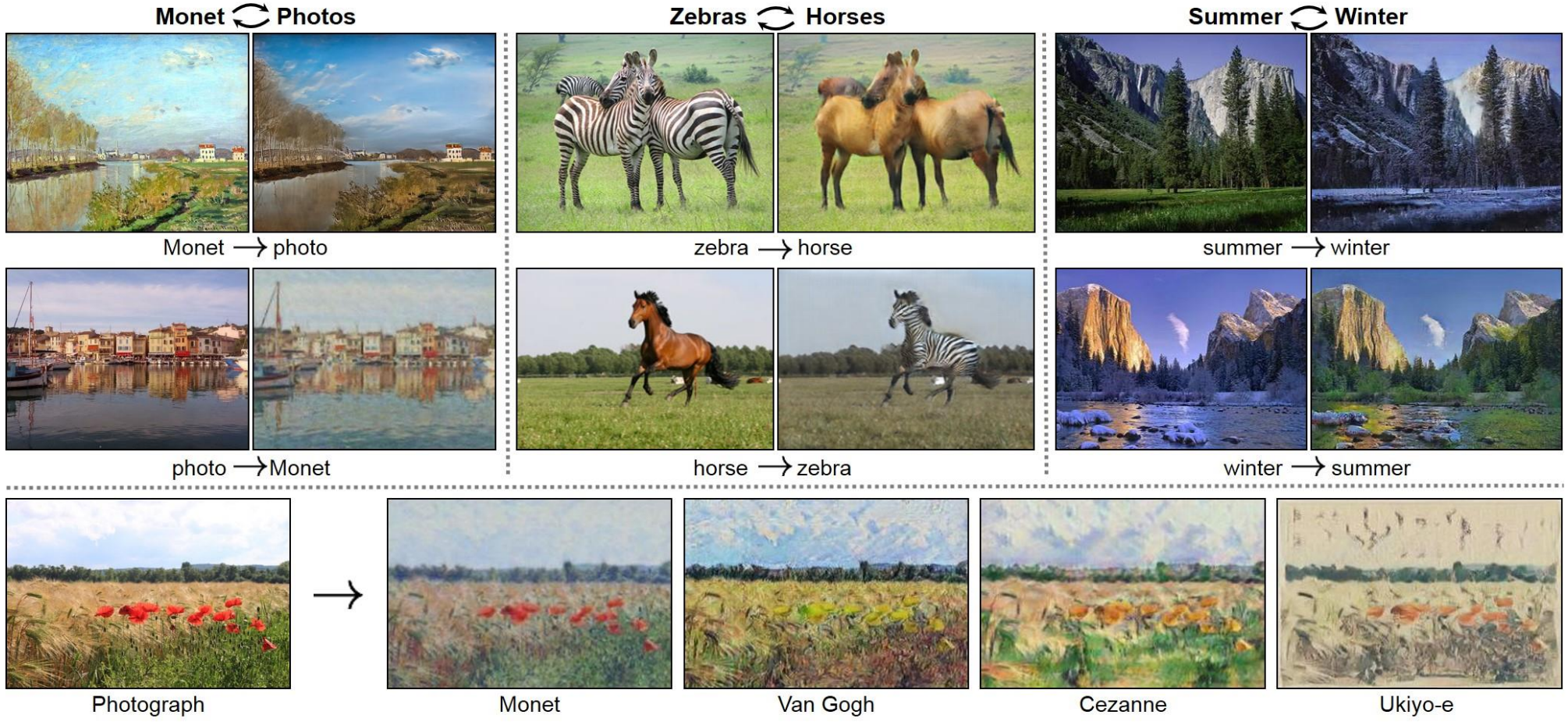
[11]

Cycle-GAN



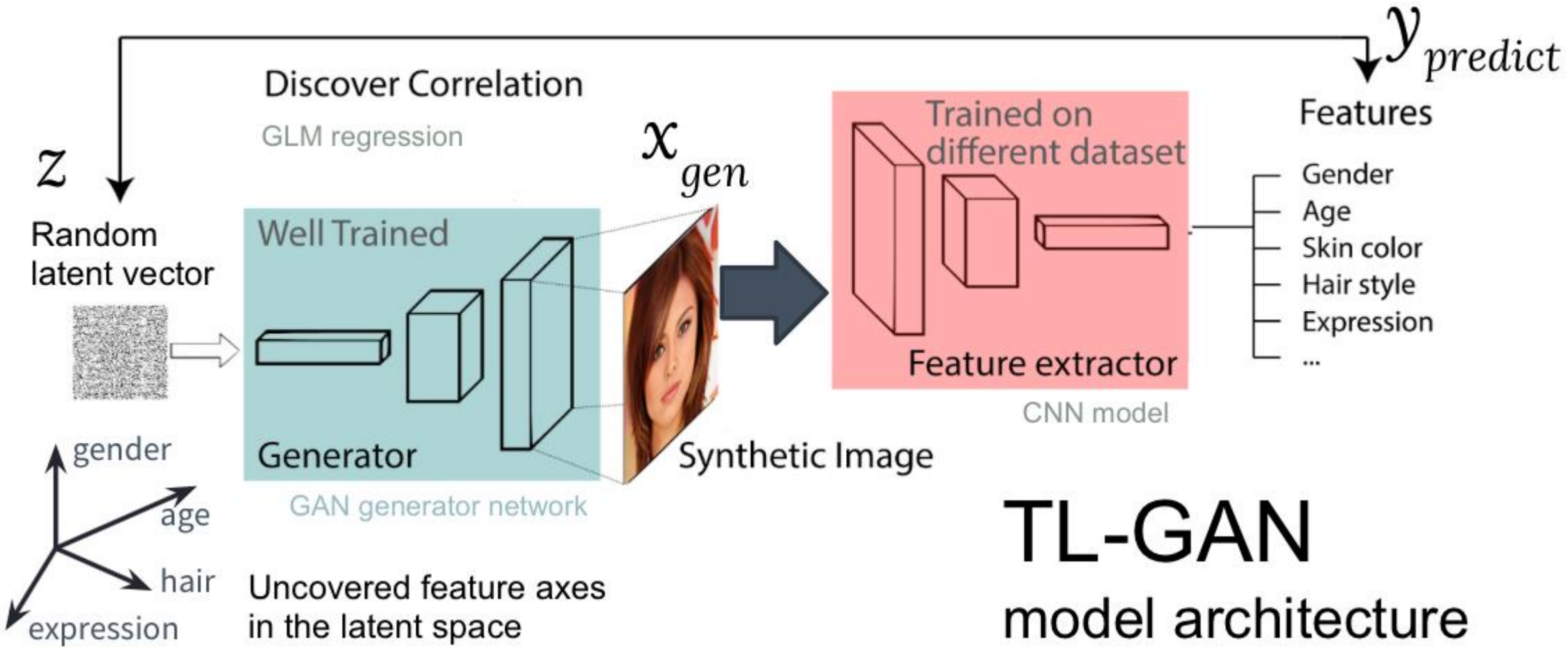
https://hardikbansal.github.io/CycleGANBlog/ [13]

Cycle-GAN

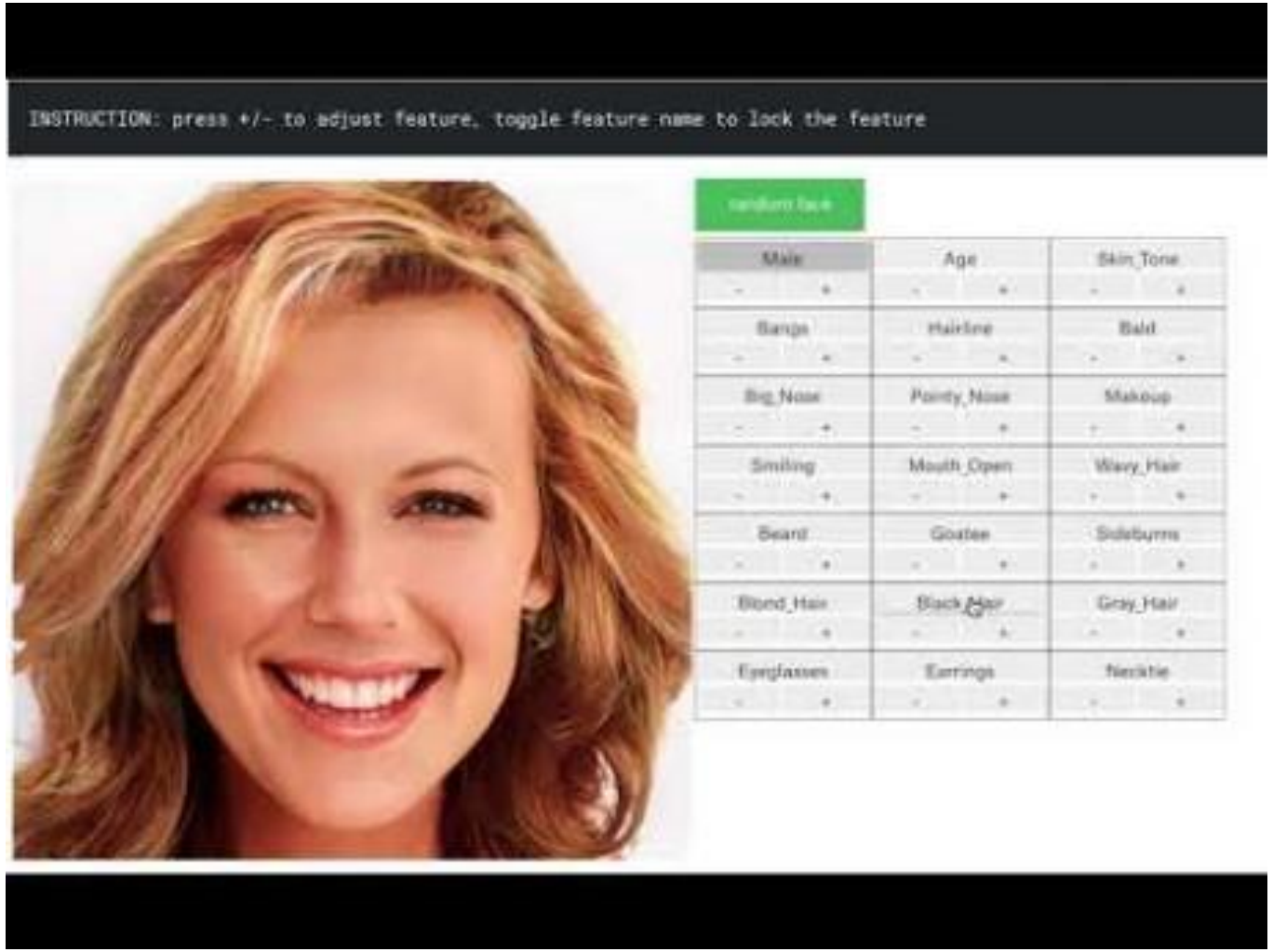

Monet ⇄ Photos
Zebras ⇄ Horses
Summer ⇄ Winter
Monet → photo
zebra → horse
summer → winter
photo → Monet
horse → zebra
winter → summer
Photograph
Monet
Van Gogh
Cezanne
Ukiyo-e

[13]

Transparent-Latent-Space-GAN

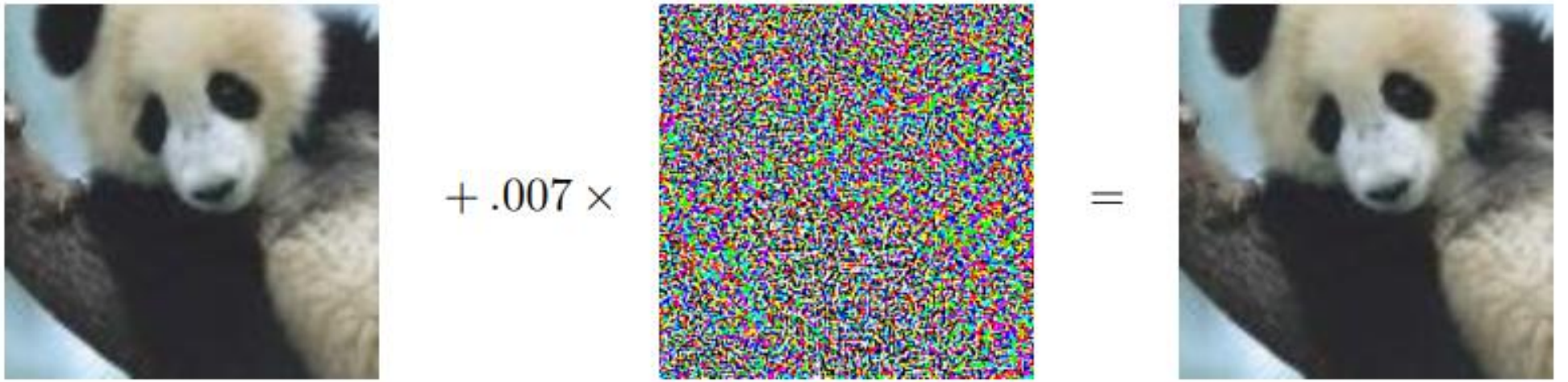


Transparent-Latent-Space-GAN



https://www.youtube.com/watch?v=O1by05eX424

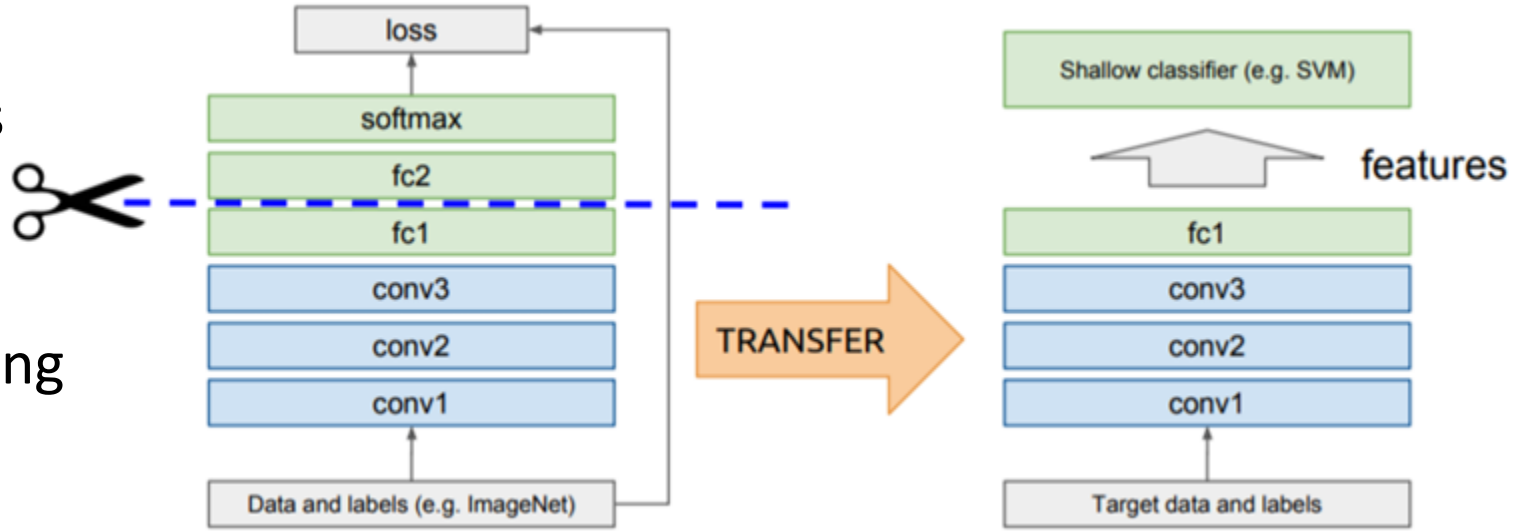
Adversarial Attacks



[5, 13]

Transfer Learning

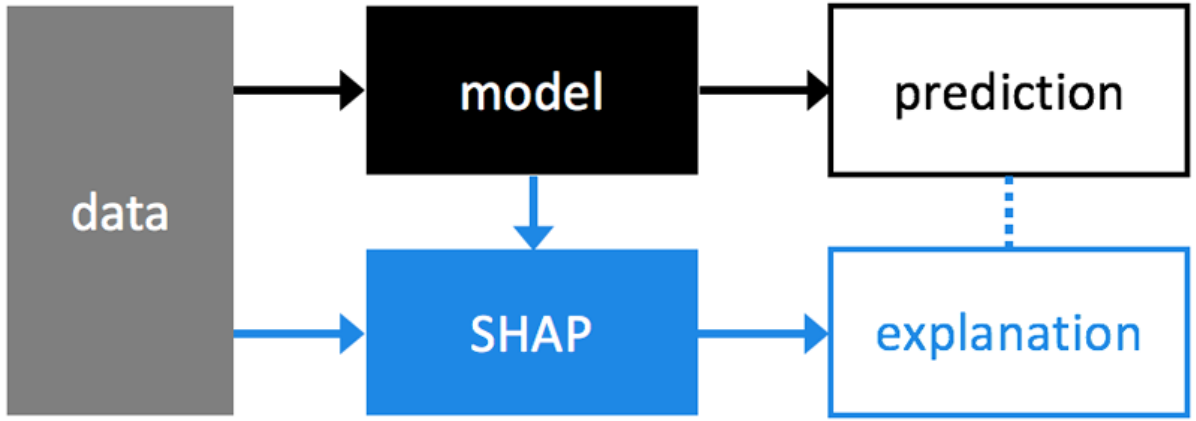
- One thing they (almost) all have in common
  - They are huge!
  - Millions of parameters
  - Need lots of data and computation / time to train
- Solution
  - Use pretrained models
  - Replace last layer(s)
  - Freeze existing layers
  - Retrain with low learning rate on your data



Interpretability

- SHAP (Shapley Additive Explanation)
  - Fit simple linear (binary) model to approximate a more complex model
  - Set of methods for different models
    - Shapely values
    - LIME
    - DeepLIFT

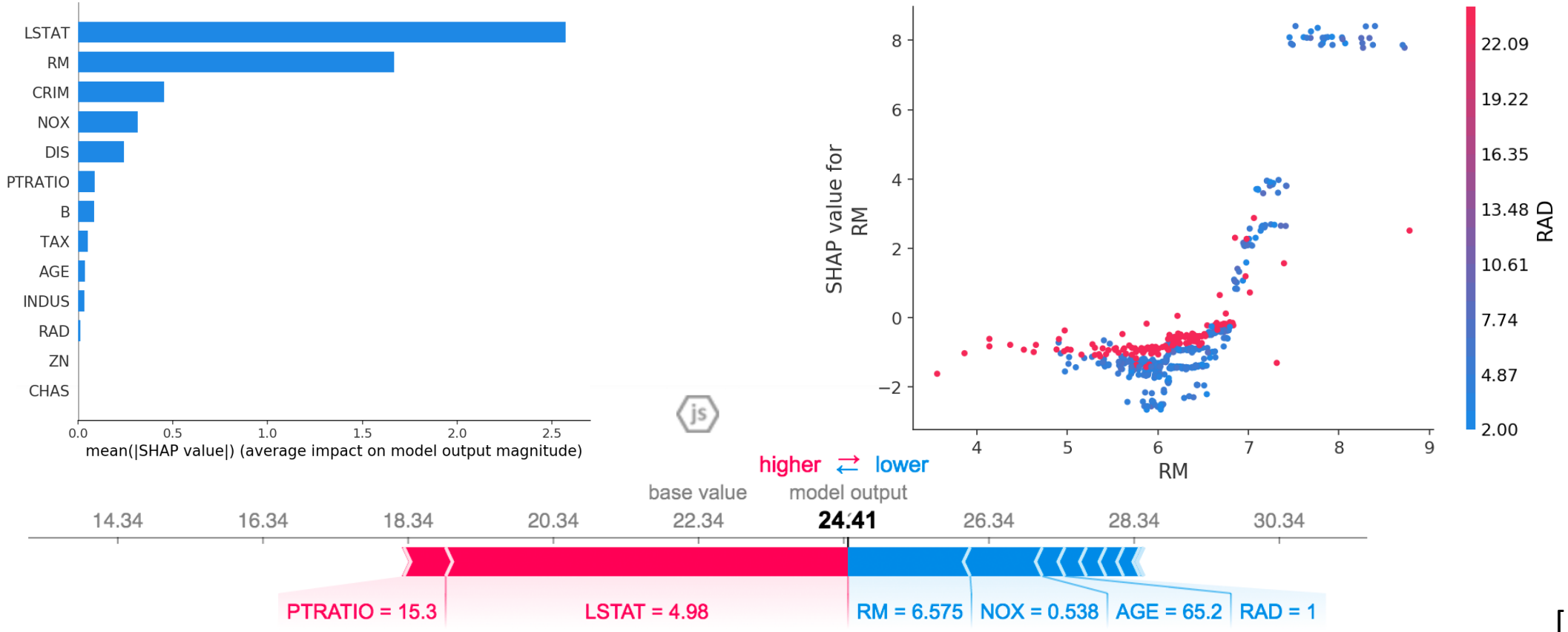
$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$



[12]

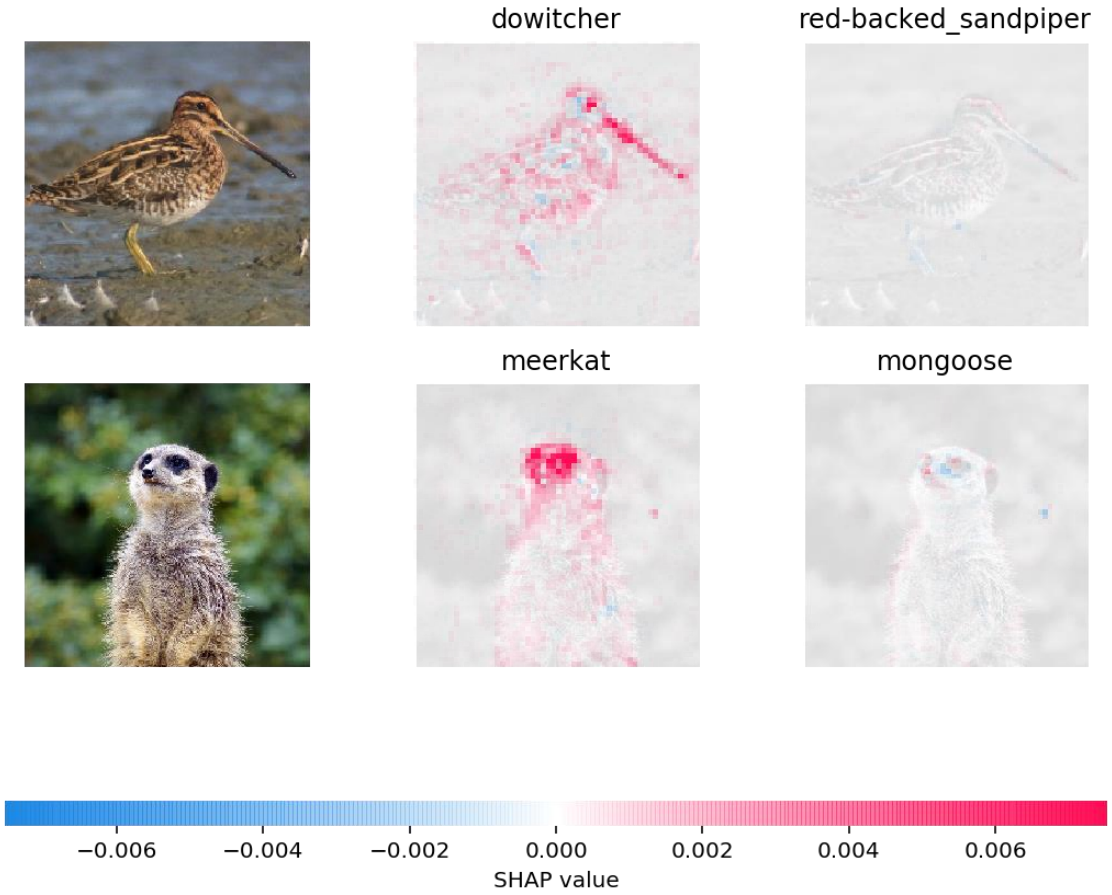
Interpretability

- SHAP (Shapley Additive Explanation)



[12]

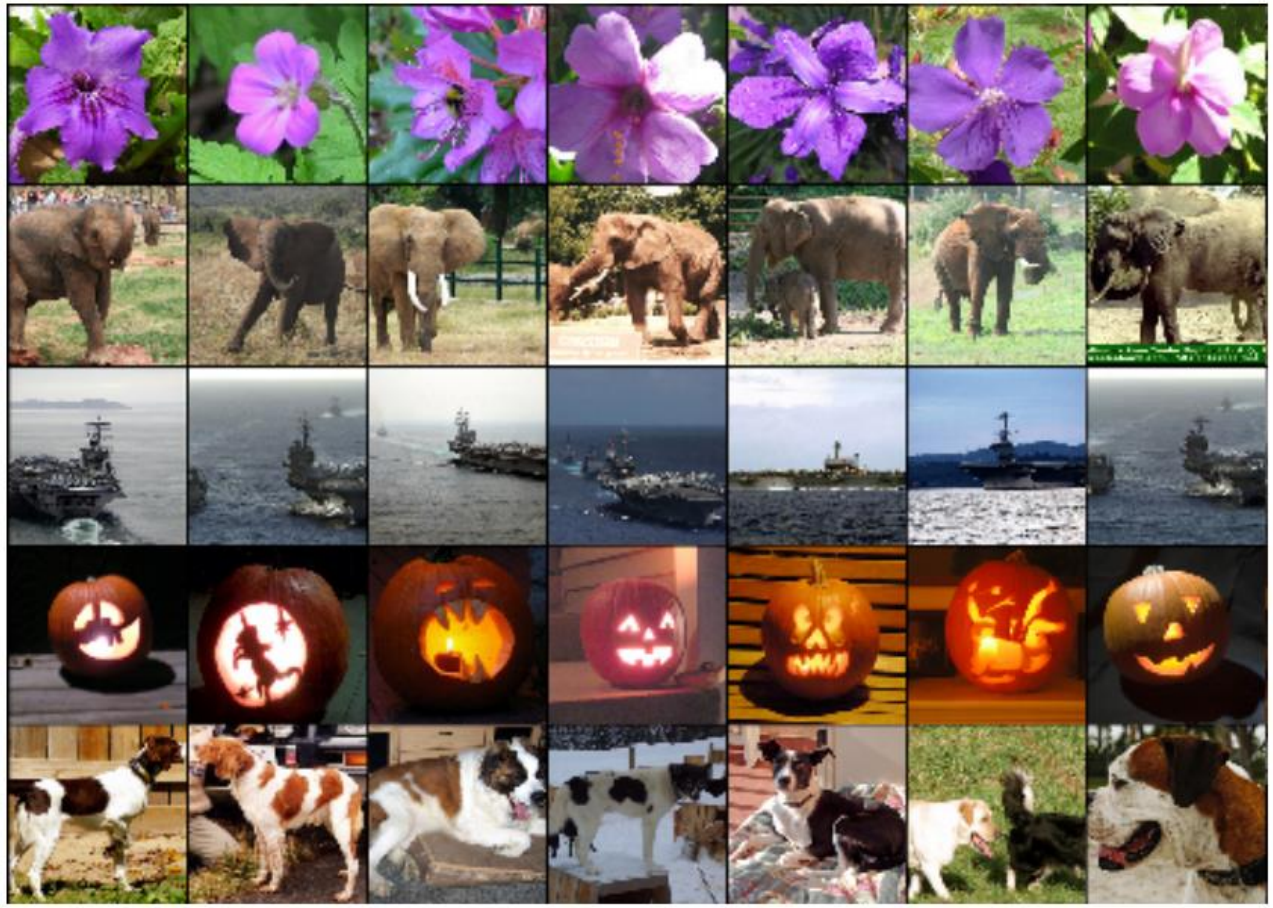
Interpretability



https://github.com/slundberg/shap

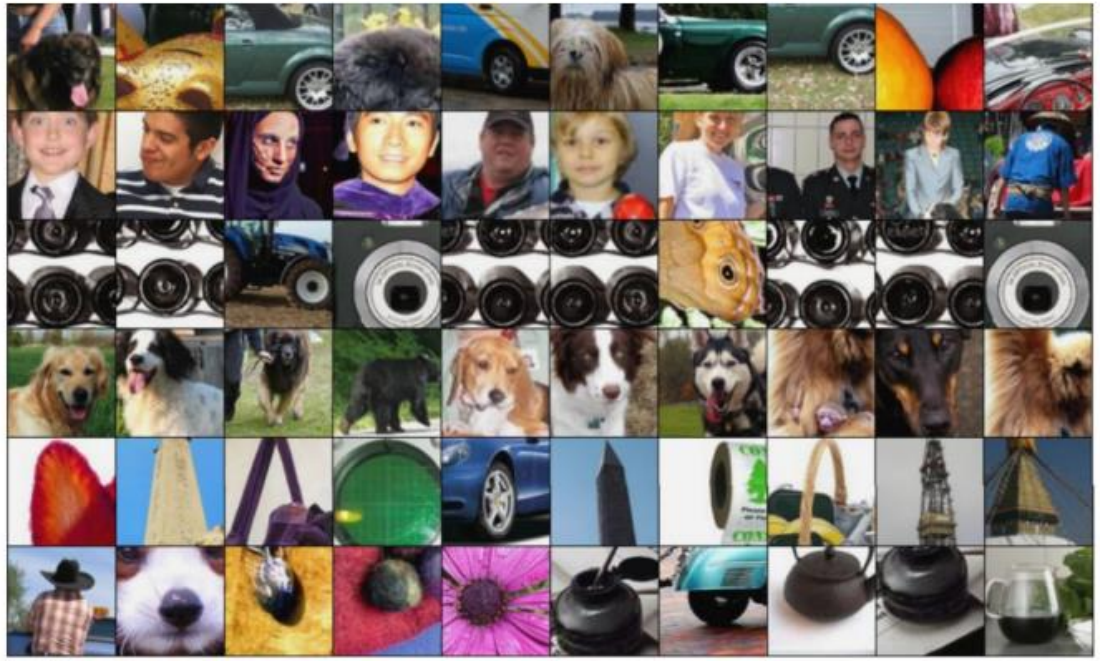
[12]

Interpretability



[10]

Interpretability



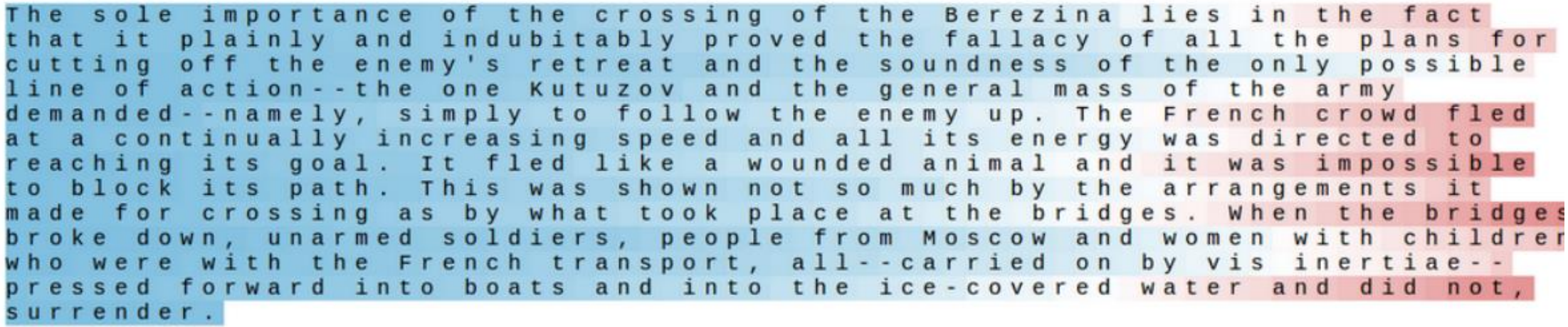
Maximally activating patches
(Each row is a different neuron)

Guided Backprop

[1, 14]

Interpretability

Cell sensitive to position in line:



Cell that turns on inside quotes:

[4]

Frameworks

- Tensorflow (Python, Javascript, C++, R, Swift, Go)
- PyTorch (Python)
- Torch (Lua, C)
- CNTK (Python, C++, C#, Java)
- Theano (Python, MATLAB, C++)
- Keras (Python, R)
  - Easy interface for Tensorflow, Theano, CNTK

References

(1) Fei-Fei Li, Justin Johnson and Serena Yeung (2019) CS231n: Convolutional Neural Networks for Visual Recognition. Stanford. http://cs231n.stanford.edu/.

(2) Canziani, A., Paszke, A., & Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

(3) Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495.

(4) Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

(5) Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

(6) Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

(7) Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R. & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.

(8) Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

(9) Spinner, T., Körner, J., Görtler, J., Deussen, O. (2019) Towards an interpretable latent space. https://thilospinner.com/towards-an-interpretable-latent-space

(10) Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

(11) Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

(12) Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (pp. 4765-4774).

(13) Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

(14) Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*.

(15) Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

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