Deep Learning - 2019

Conclusions

Prof. Avishek Anand

Exam Prep

- Exam Duration : 2 hours
- Date: 8th August
- Max marks : 100

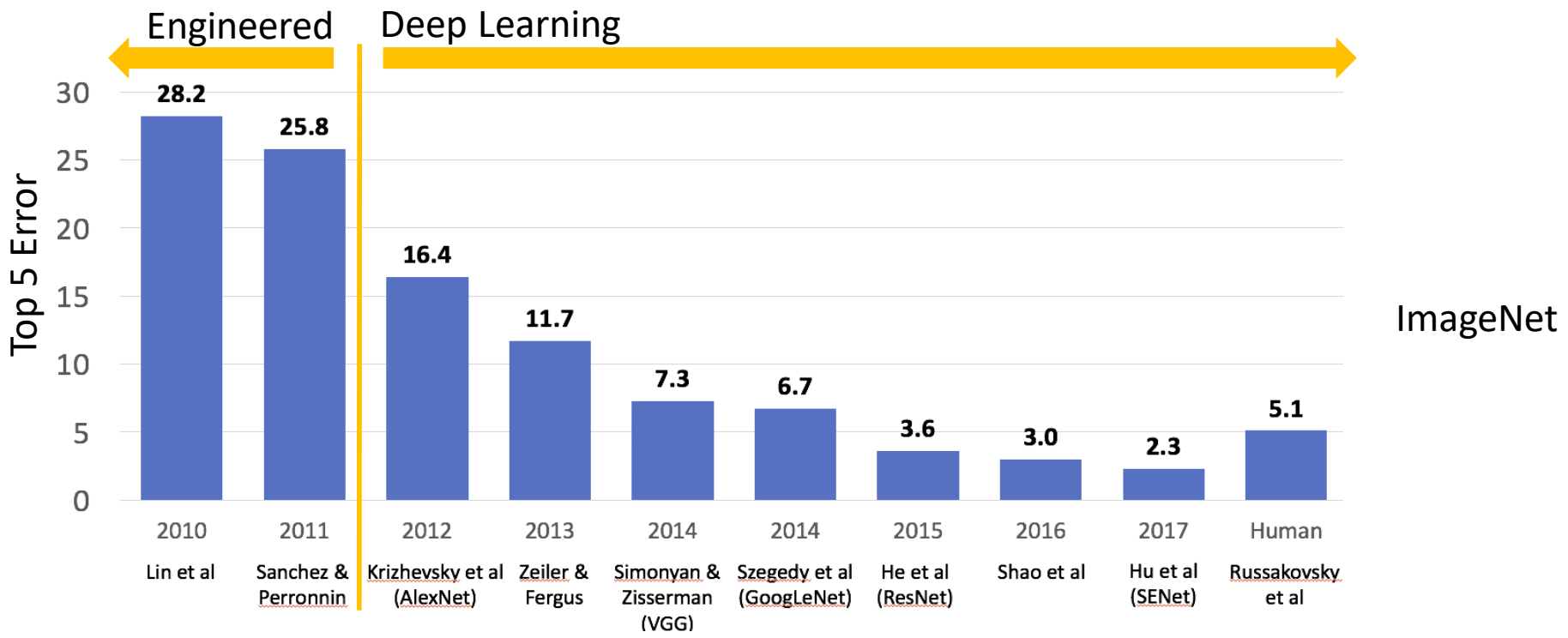
- Calculators allowed
- 5 questions of 25 points each
  - We will take the best of four

- Written exam = 75% of the credit
- Project = 25% of the credit

- Read the instructions carefully..

Deep Learning as Representation Learning

- New pipeline advertised by DL:
  - input data -> learn representation -> Build Models
  - learn representation together with classification **(end-to-end training)**

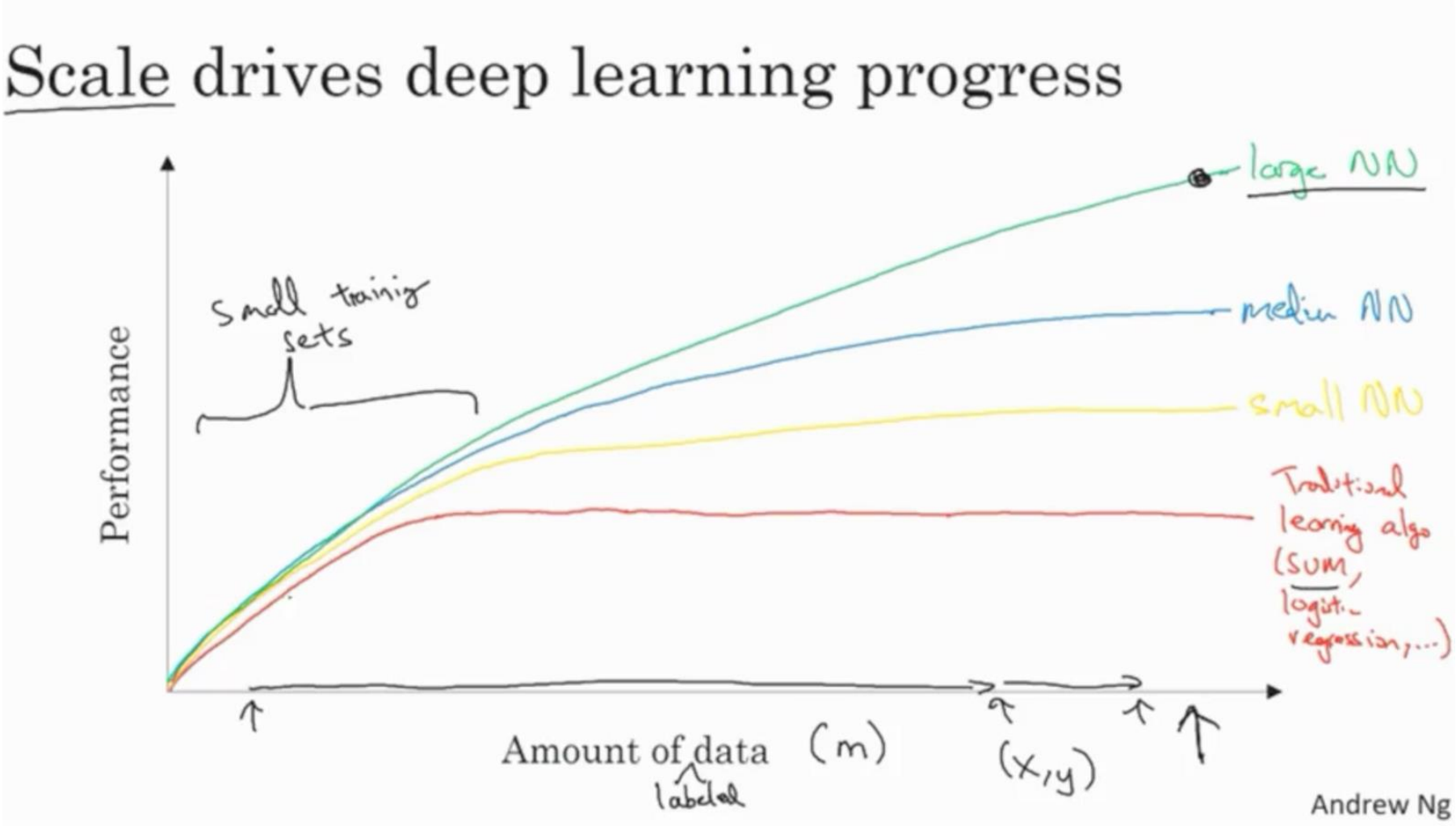


DL as Representation Learning

- A lot of time spent on creating new architectures
  - Still better performance than engineering features
  - Still less time than engineering features
  - there is still hope that with progress we will need less and less engineering of the architectures, and this will be either automated or we find the great ones

- Great progress on problems with unclear engineered features
  - Multimodal representation -- how to model language and visual signals at once ?

Why Deep Learning ? (revisited)

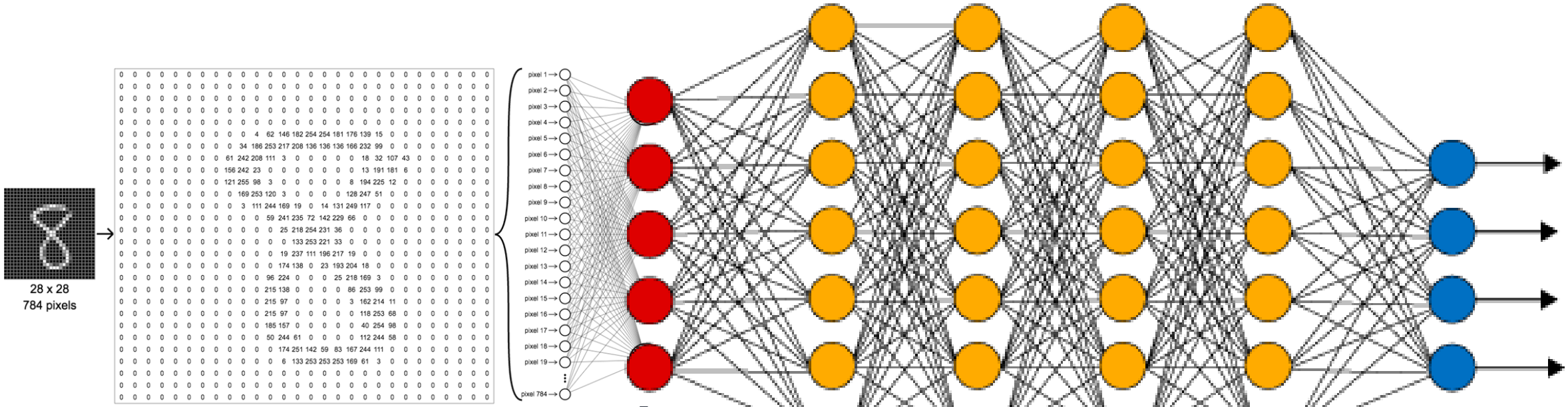


Basic Notions and Intuitions

- Joint training
  - Training at least two modules together
  - Pre-DL era: hand-engineer all the modules, train only the last one (e.g., classifier)
- End-to-end training (e2e training)
  - Join training of all the modules
- Back-propagation (Backprop)
  - Technique for training Neural Networks (compositions of functions)
  - Technically it computes a Jacobian matrix by using a chain rule
  - But it does it efficiently
- Optimization
  - We often try to find the best parameters that 'explains data' under the current model (e.g., neural network)
  - We did it in the curve-fitting example
  - There are, however, other alternatives like sampling

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w})$$

Neural Network as composition of functions

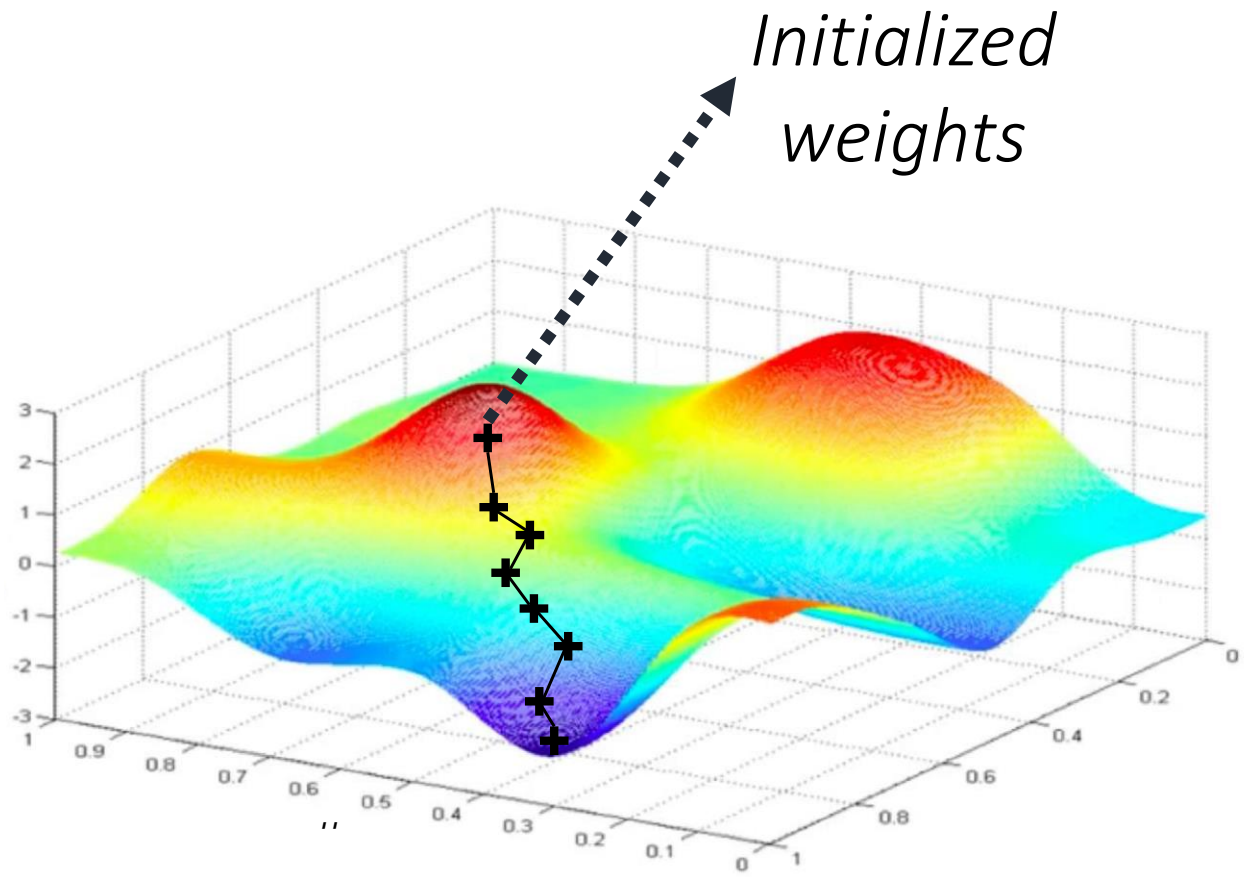


$$\mathbf{z}_1 = g\left(\mathbf{W}_1^T \mathbf{x}\right)$$

$$\mathbf{z}_2 = g\left(\mathbf{W}_2^T \mathbf{z}_1\right)$$

$$\mathbf{z}_k = g\left(\mathbf{W}_k^T \mathbf{z}_{k-1}\right)$$

Gradient Descent



Gradient Descent Algorithm

Initialize weights randomly $\sim \mathcal{N}\left(0, \sigma^2\right)$

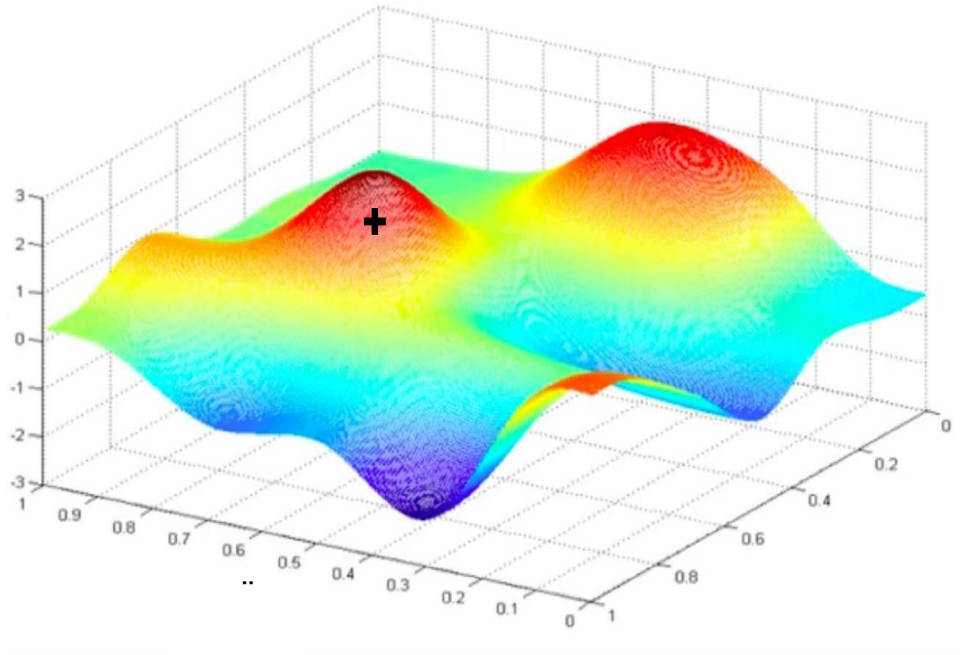
while *training error has not converged* **do**

  Compute gradient $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$

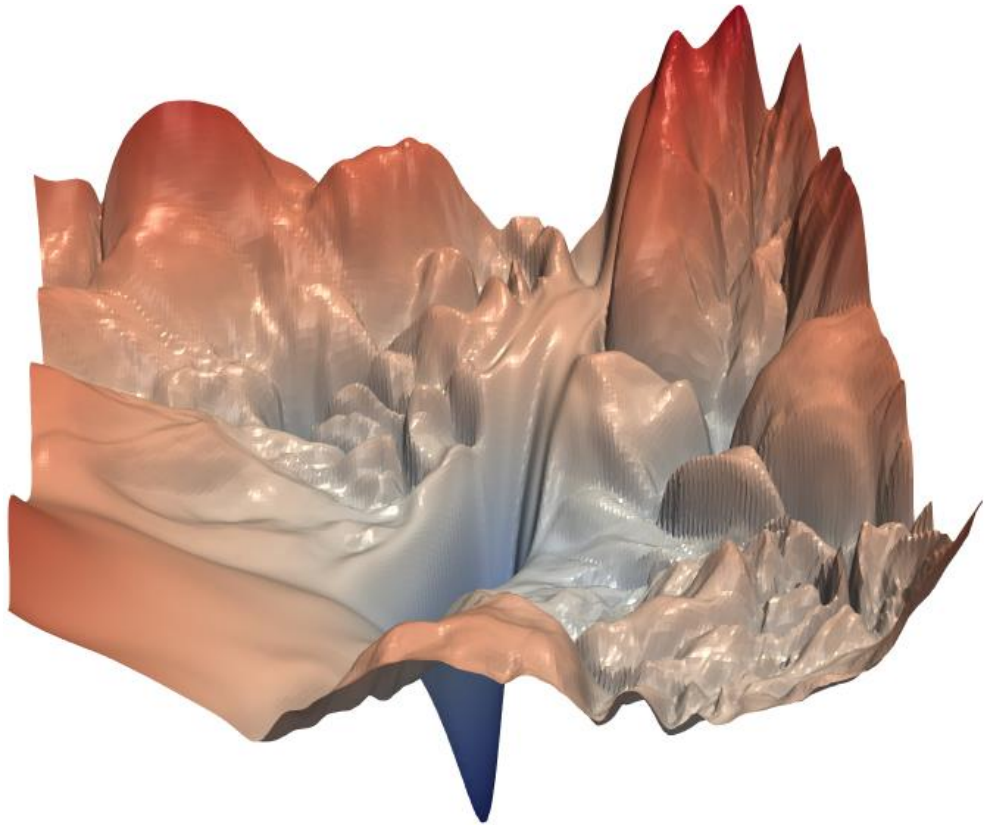
  Update weights $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$

return weights

Loss surface in reality



$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$$



Local minimas, saddle points

[Visualizing the loss landscape of neural nets, Li et al 2018]

Optimization Techniques: Learning rate

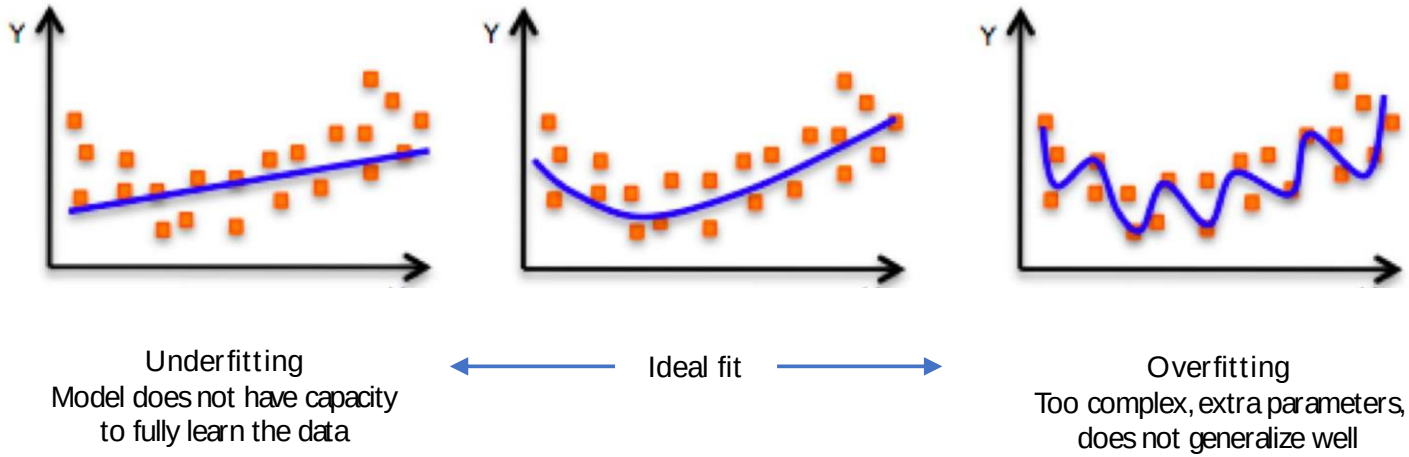
- Small learning rates have slow convergence and get stuck in local minimas
- Large learning rates overshoot and might diverge
- Stable learning rates converge smoothly and avoid local minima

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$$

- Fix learning rates based on
  - The size of the gradient
  - Rate of learning
  - Size of particular weights
  - … <More on this in upcoming lectures>

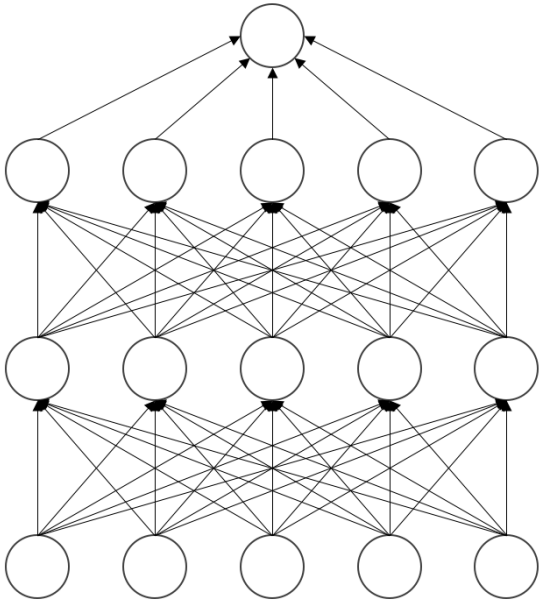
Momentum
Adagrad
Adadelta
Adam
RMSProp

Regularization Techniques

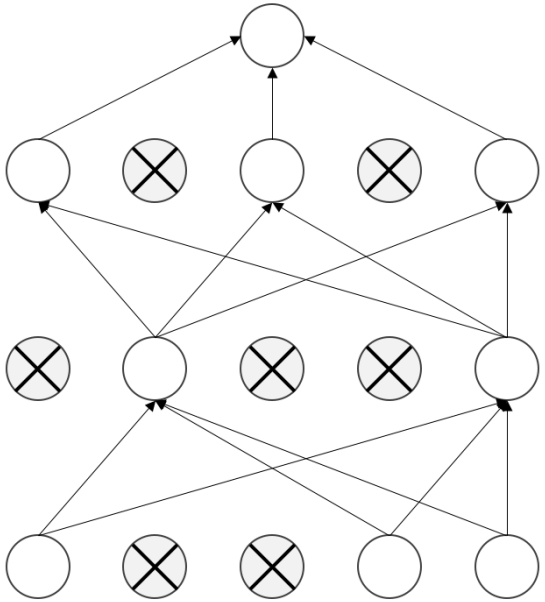


- Deep Neural Networks have a large capacity (leads to more complex models) and hence tend to overfit
- Regularization: Forces the model to learn simpler model
- How do we regularize deep networks ?

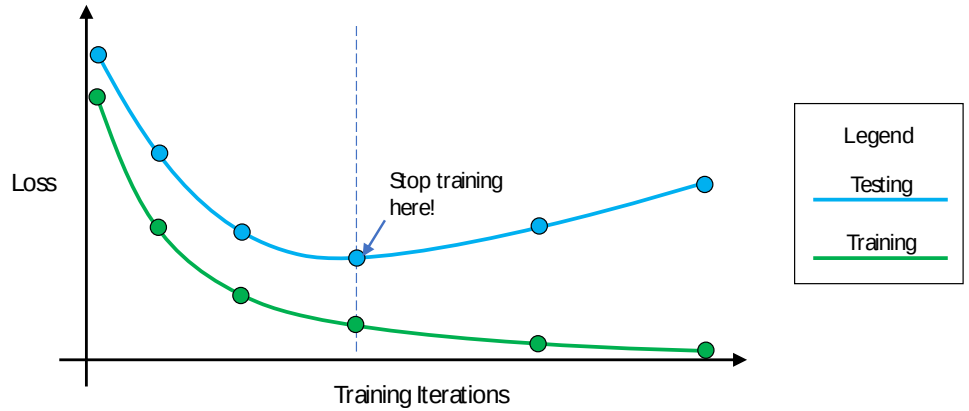
Dropout and Early Stopping



Standard Neural Net

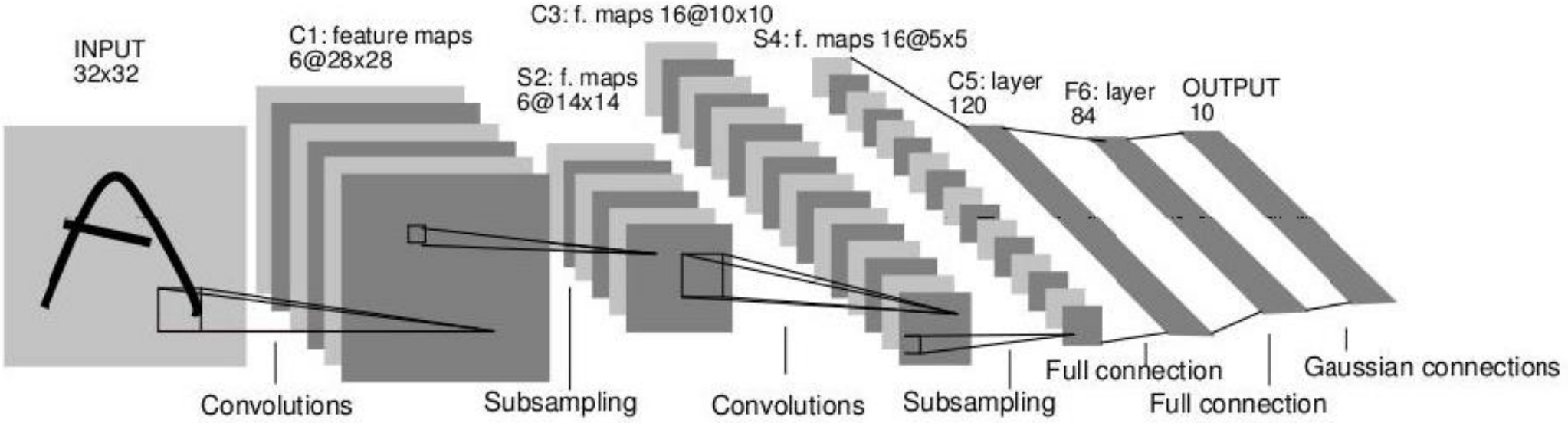


After applying dropout



- Drop some of the neurons (setting activation = 0)
- Forces the network to learn using a smaller capacity --> robust to overfitting

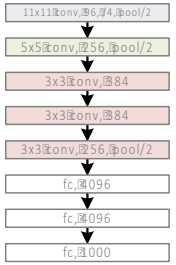
Convolutional Neural Nets



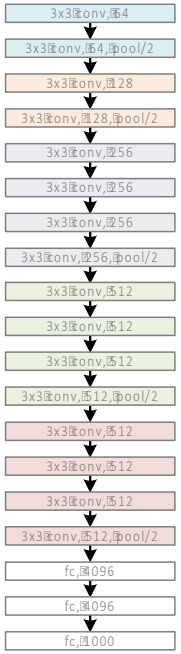
- Filters, Strides, Pooling, Feature Maps

Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)

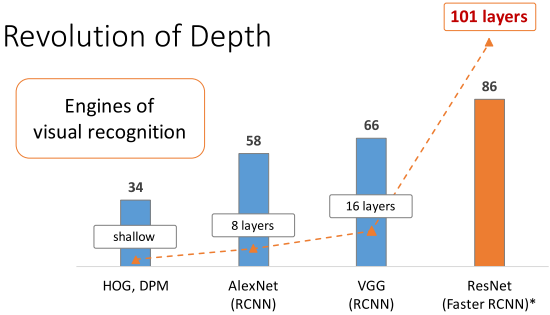


VGG, 19 layers (ILSVRC 2014)



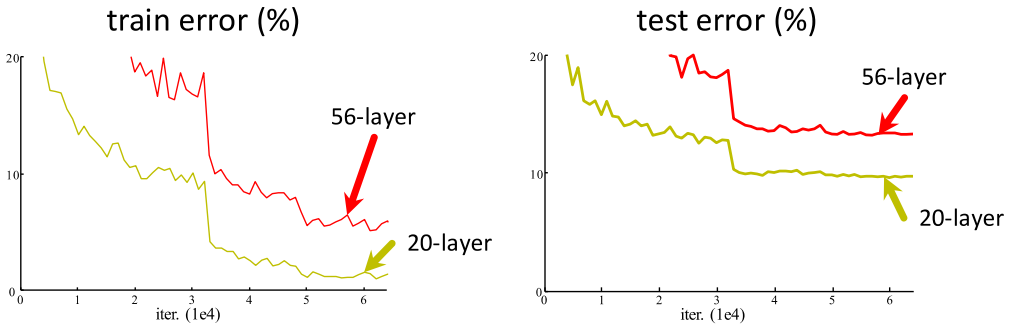
GoogleNet, 22 layers (ILSVRC 2014)

ResNet, 152 layers (ILSVRC 2015)



But is learning as simple as stacking more and more layers ?

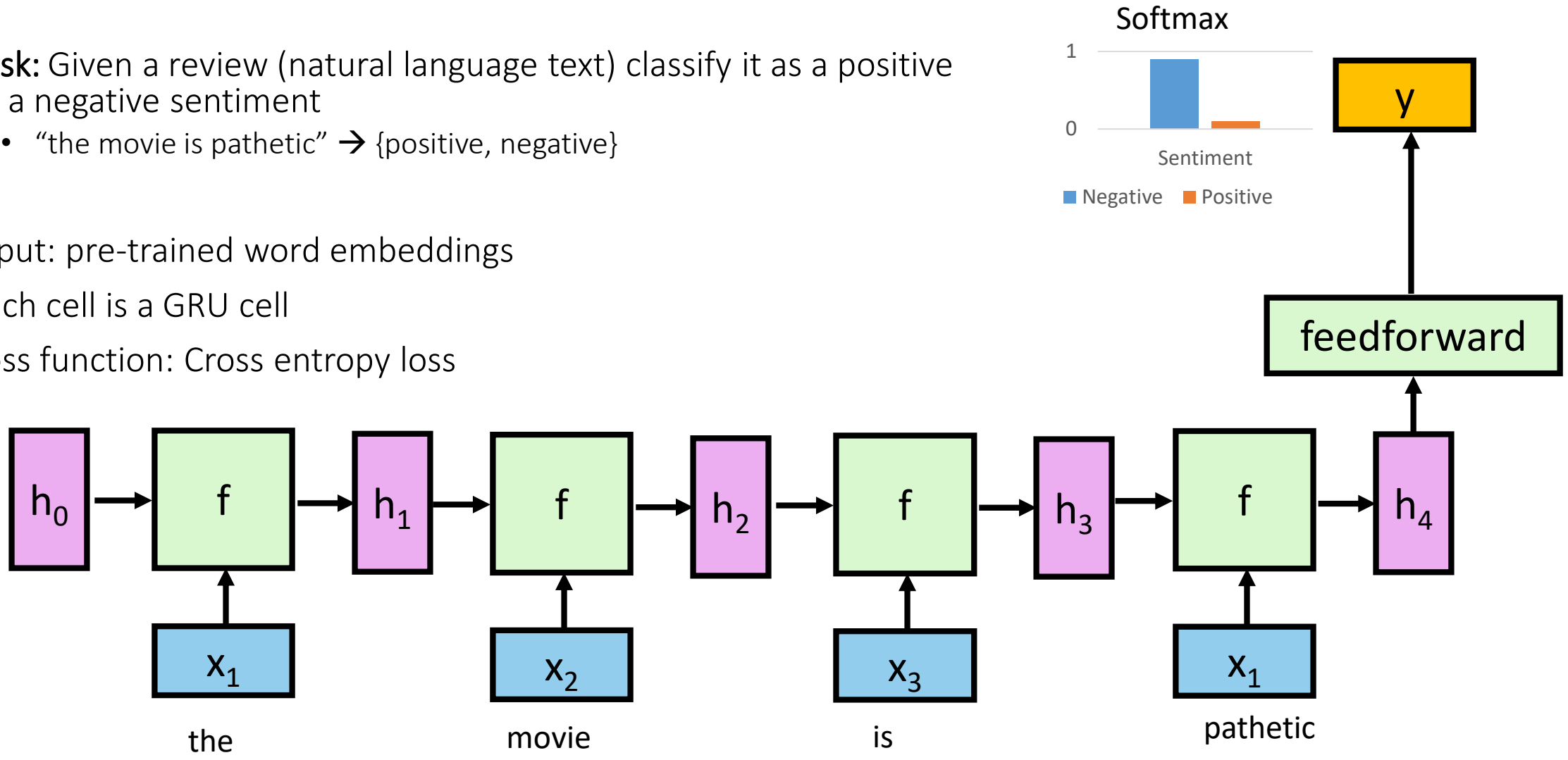
CIFAR-10



RNN for Sequential Data

- **Task:** Given a review (natural language text) classify it as a positive or a negative sentiment
  - "the movie is pathetic" → {positive, negative}

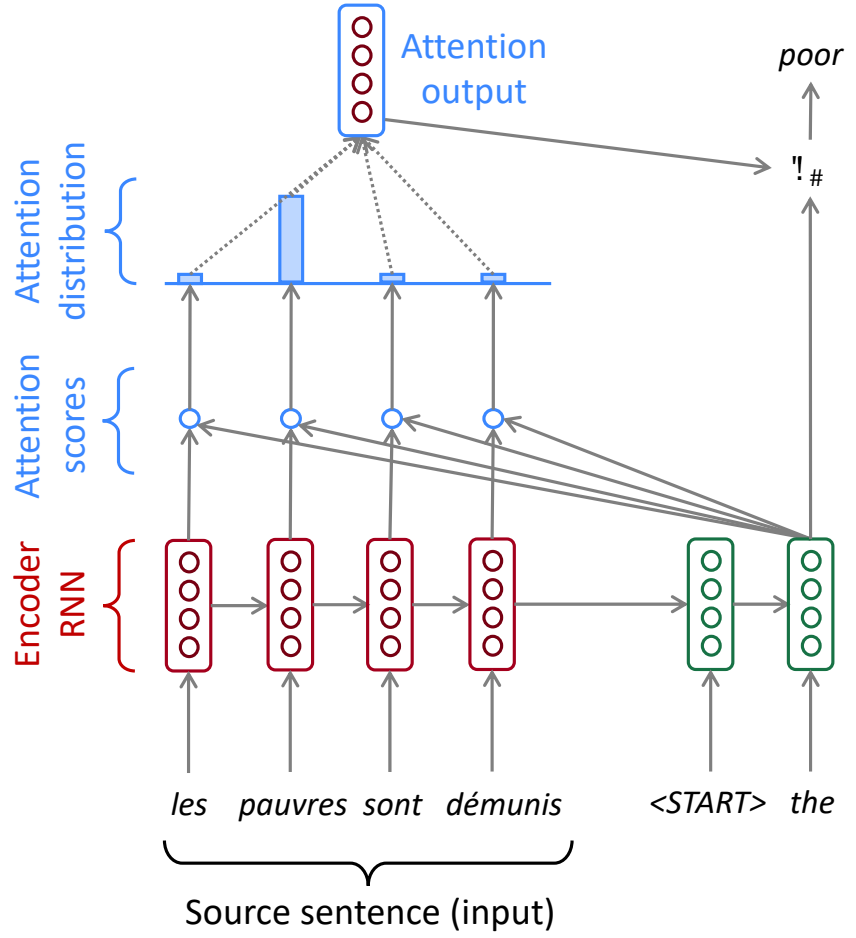
- Input: pre-trained word embeddings
- Each cell is a GRU cell
- Loss function: Cross entropy loss



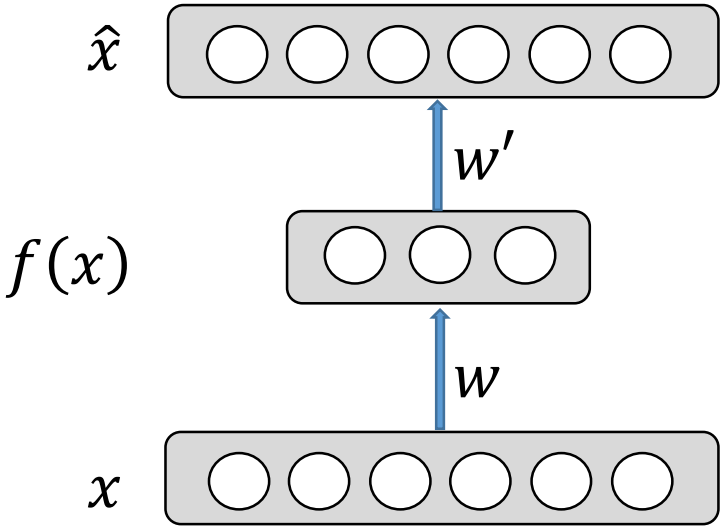
Attention Mechanism

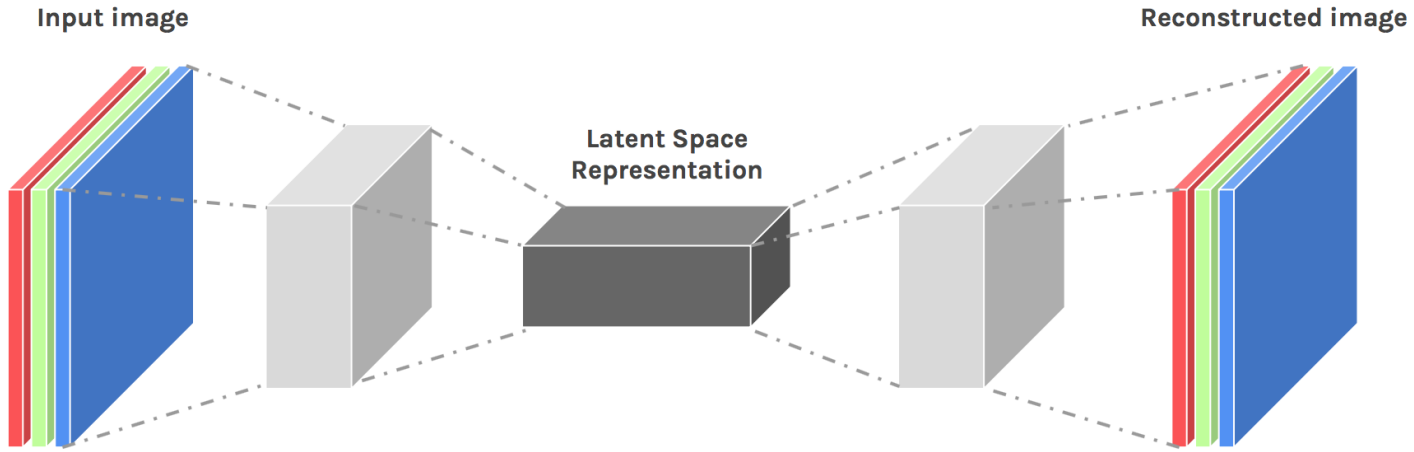
$$\alpha_i \in [0, 1]$$

$$\alpha_i \in \{0, 1\}$$



Unsupervised Methods





Convolutional AE

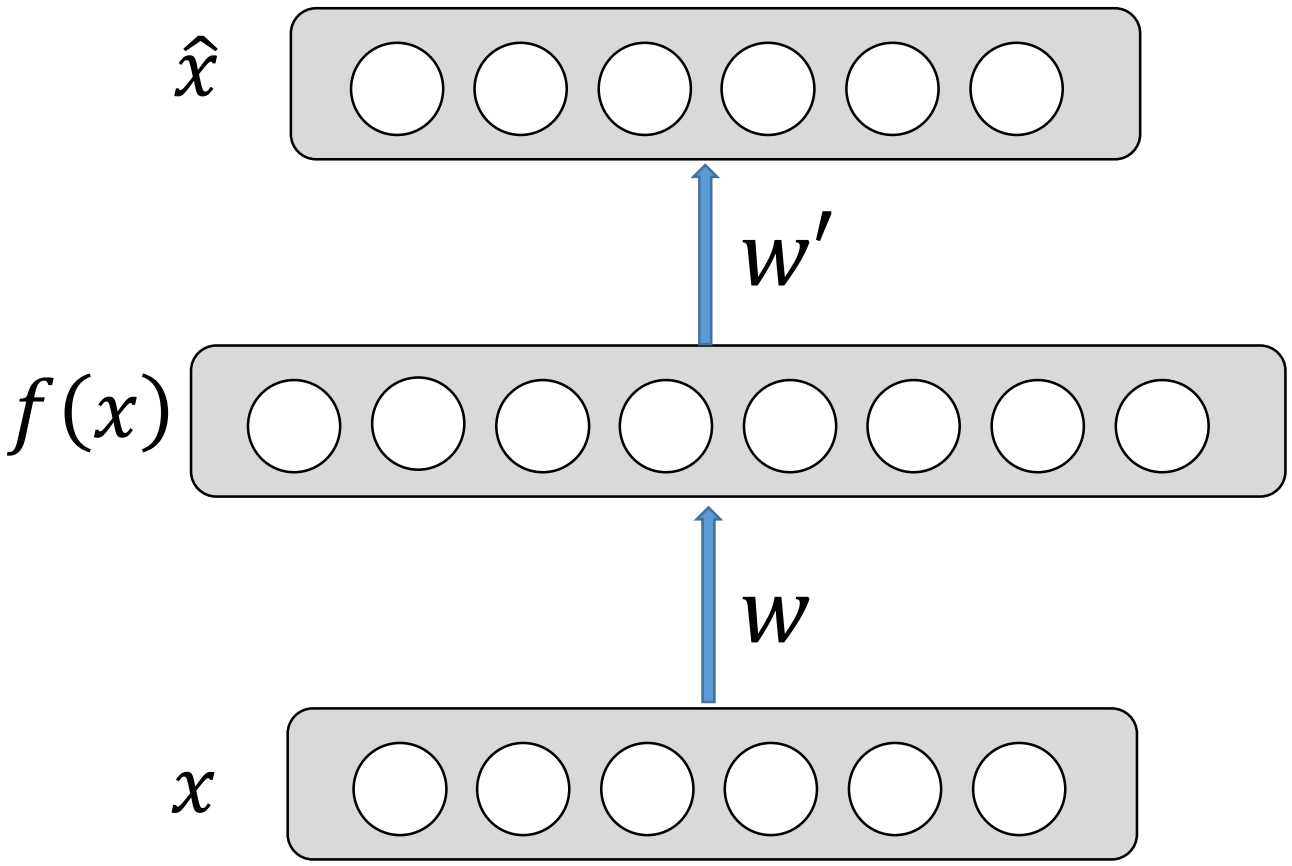
We are forcing the hidden layer to learn a generalized structure of the data

Overcomplete Auto-encoders

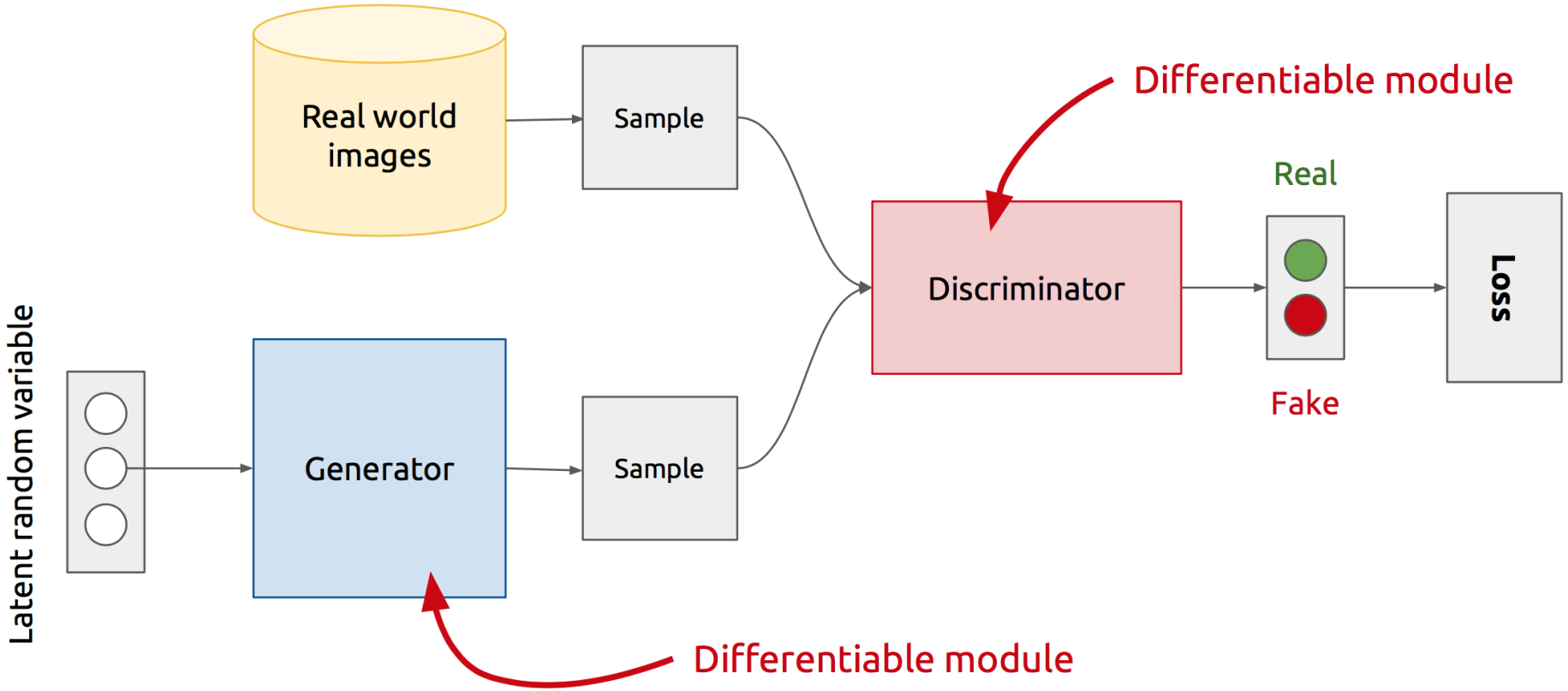
- Hidden layer is **Overcomplete** if greater than the input layer
  - No compression in hidden layer.
  - Each hidden unit could copy a different input component.

- No guarantee that the hidden units will extract meaningful structure.

- Adding dimensions is good for training a linear classifier (XOR case example).
- A higher dimension code helps model a more complex distribution.



Generative Adversarial Networks



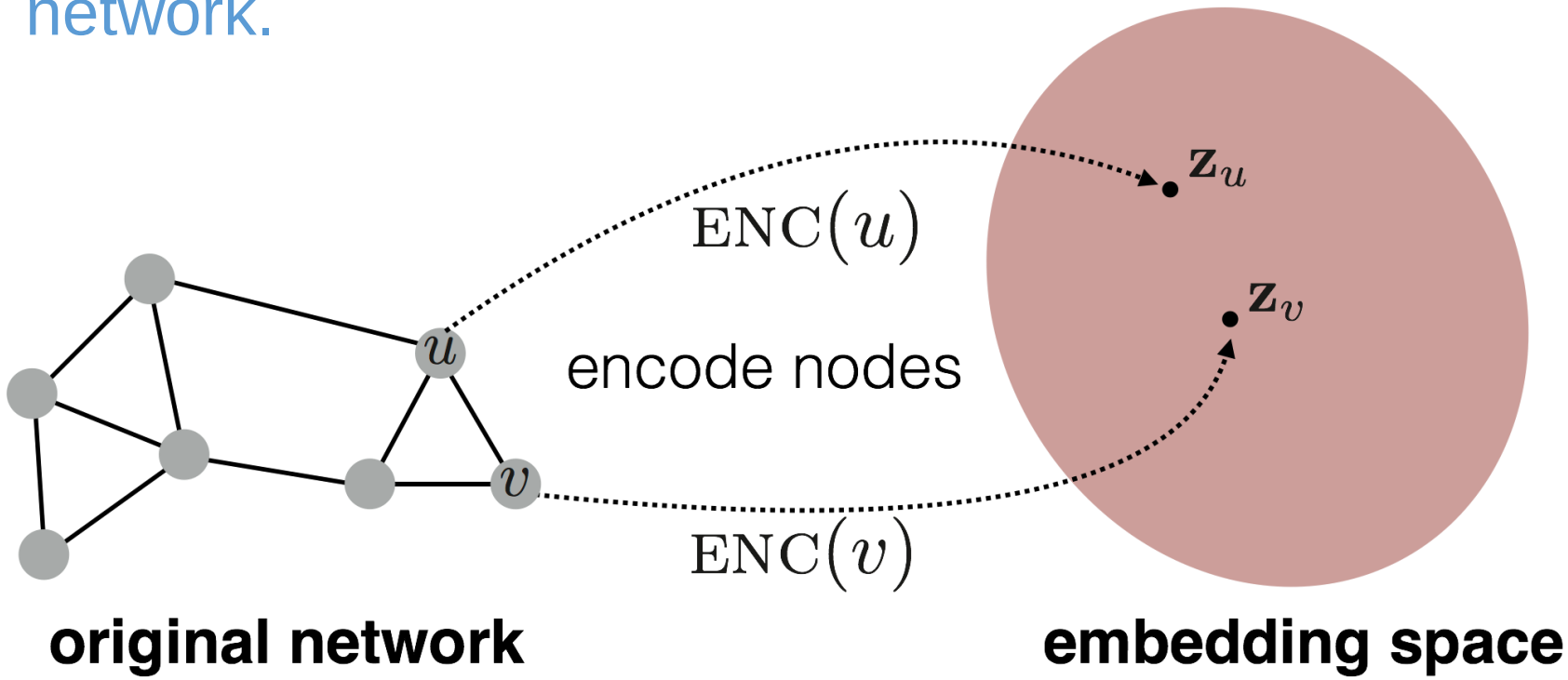
$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Discriminator output for real data x

Discriminator output for generated fake data G(z)

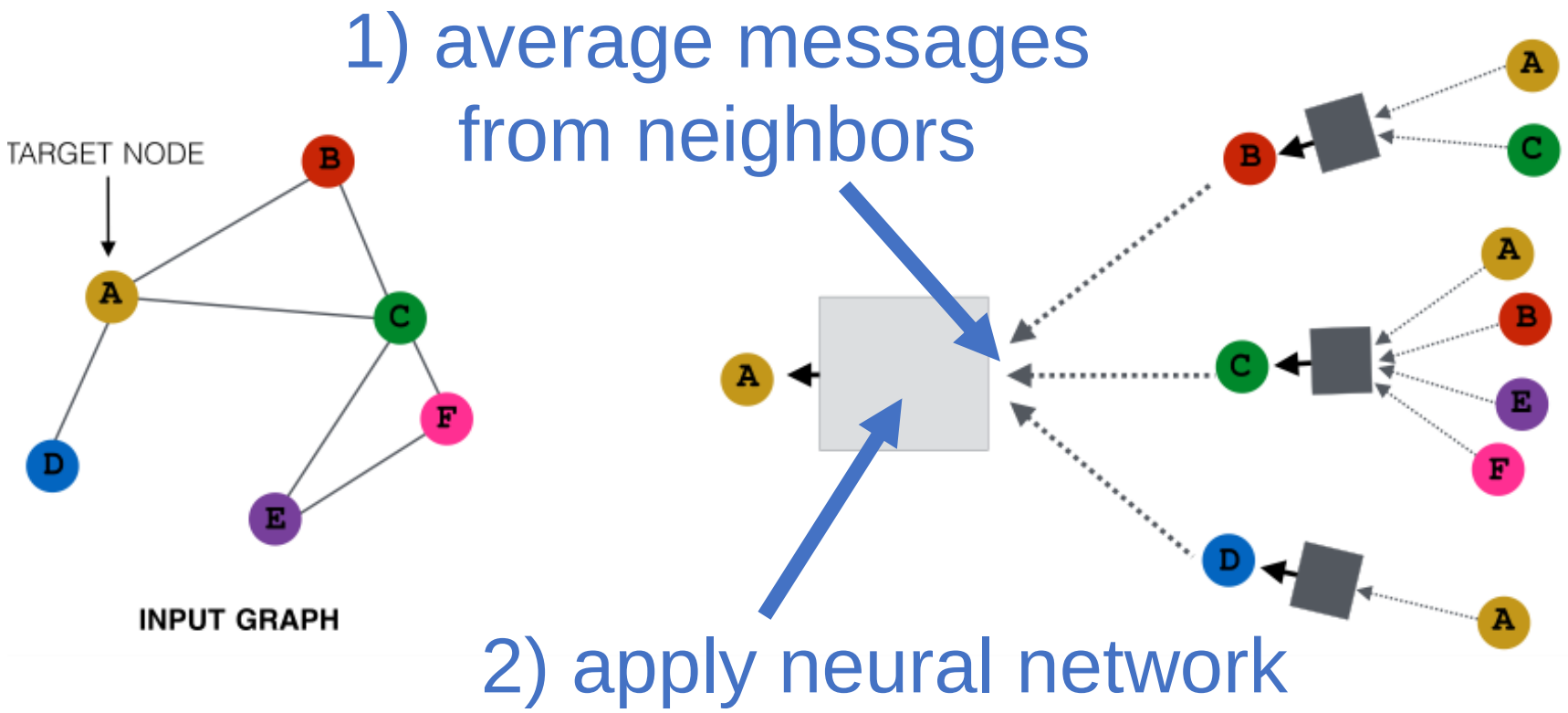
Deep Learning in Graphs

- Learning representations for nodes is to encode nodes so that similarity in the embedding space (e.g., dot product) approximates similarity in the original network.

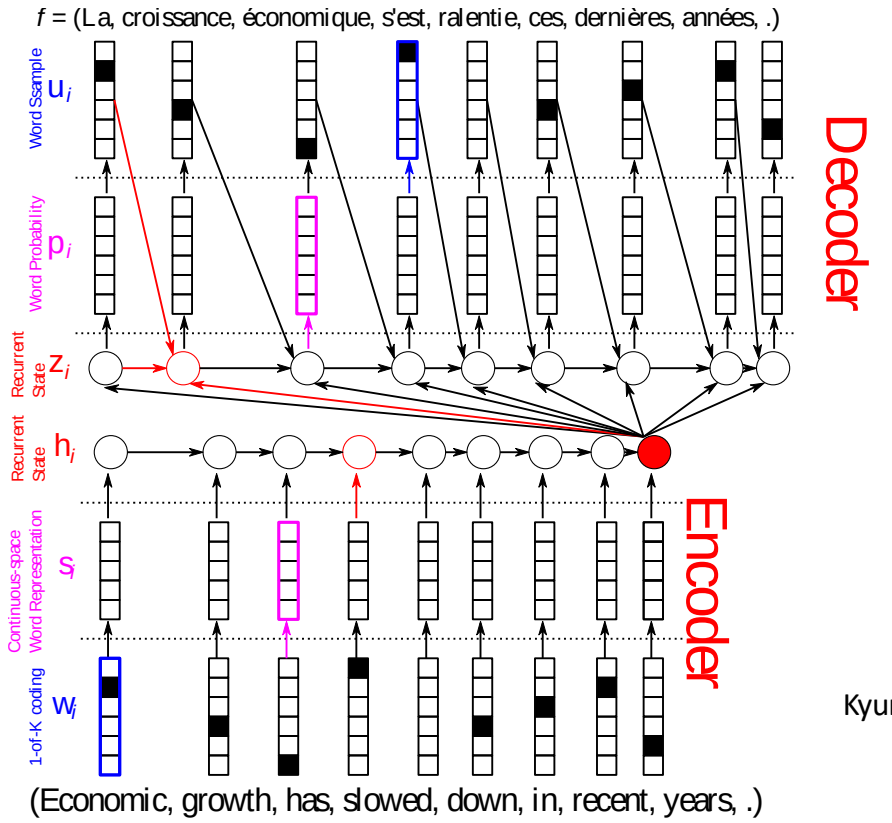


Deep Learning in Graphs

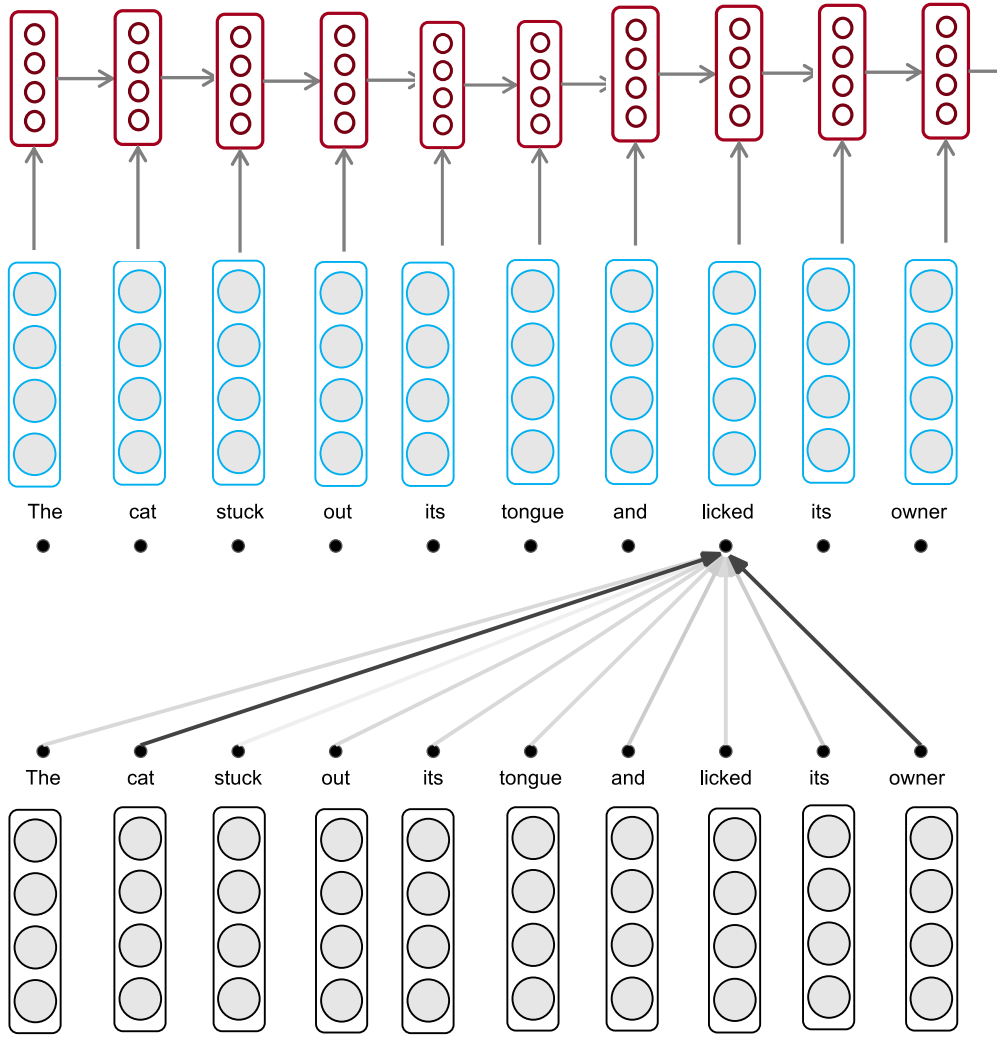
- Basic approach: Average neighbor information and apply a neural network.



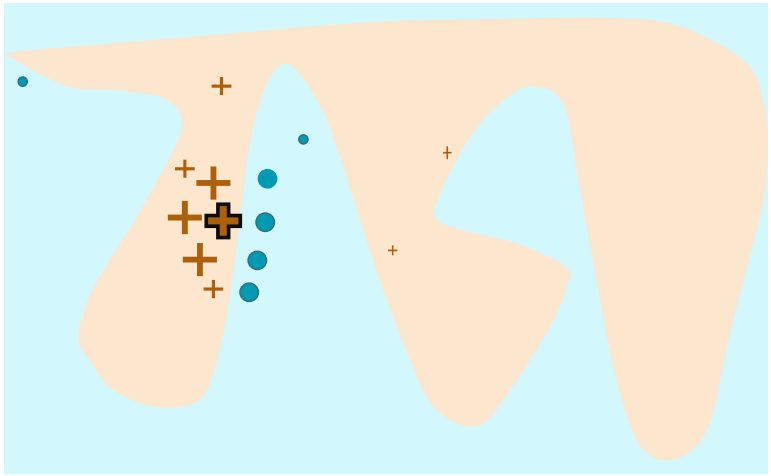
Deep Learning in Text

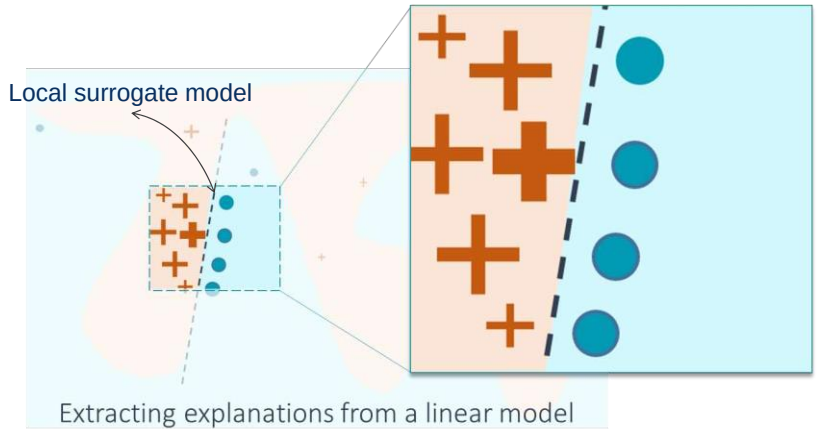


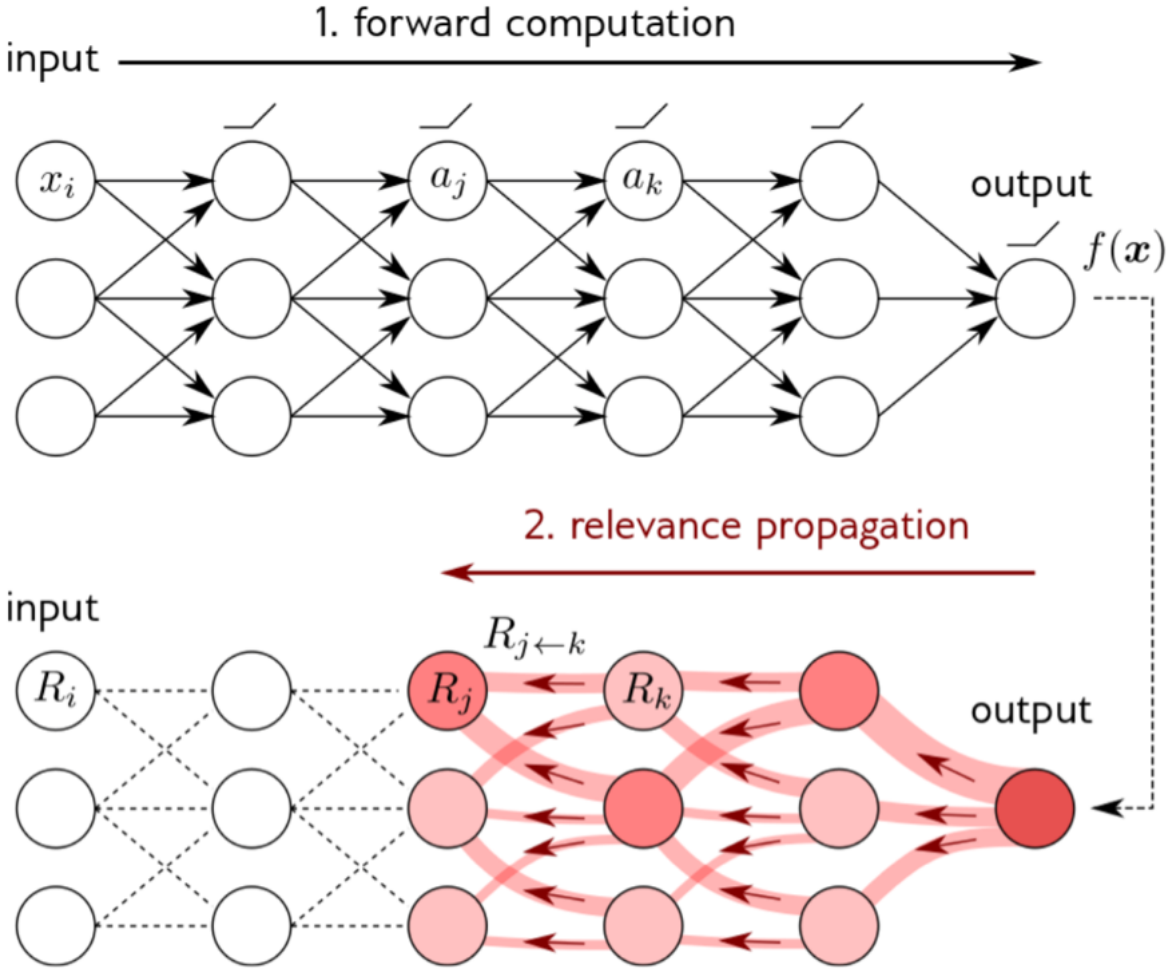
Kyunghyun Cho et al. 2014



Interpretability







Exam Prep

- Exam Duration : 2 hours
- Date: 8th August
- Max marks : 100

- Calculators allowed
- 5 questions of 25 points each
  - We will take the best of four

- Written exam = 75% of the credit
- Project = 25% of the credit

- 2 successful presentations = 0.33 grade improvement

Master/Bachelor Thesis Possibility

- We work a lot in the areas of
  - Deep Learning for Graphs
  - Deep Learning for Question Answering
  - Interpretability of Deep Learning Approaches

- Get in touch if you are motivated and have good foundations from this lecture

Thank You