Deep Learning

Karel Horák

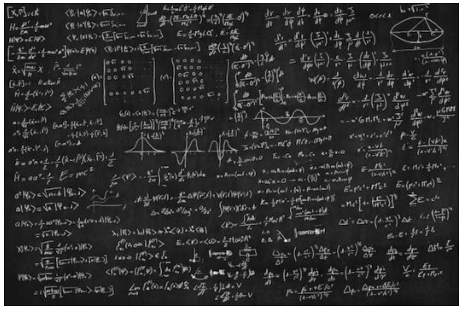
Disclaimer

- I have little experience with neural networks
- I have nearly no experience with deep learning

- My opinions (might be wrong)
- Please do tolerate (and correct ;)) inaccuracies

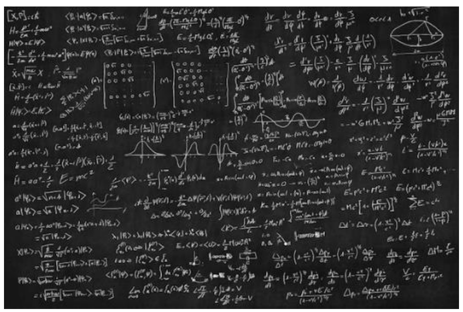
Neural Networks

Neural Networks



Math

Neural Networks



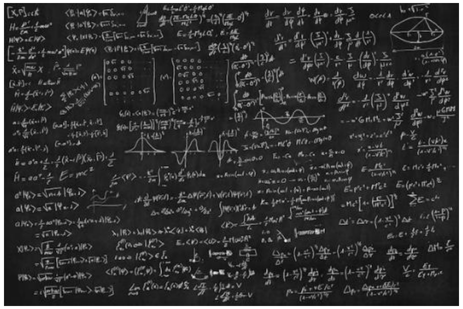
Math

+

Magic

=

Neural Networks



Math

+

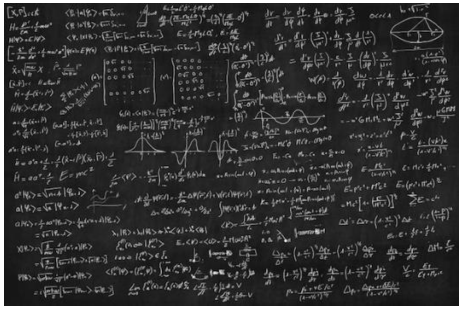
Magic

=

roundworm

brown root
rot fungus

Neural Networks



Math

5%

+

Magic

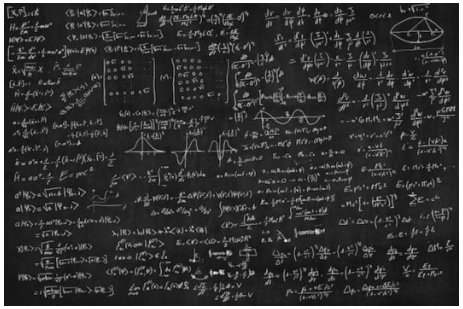
95%

=

roundworm

brown root
rot fungus

Neural Networks



Math

+

Magic

=

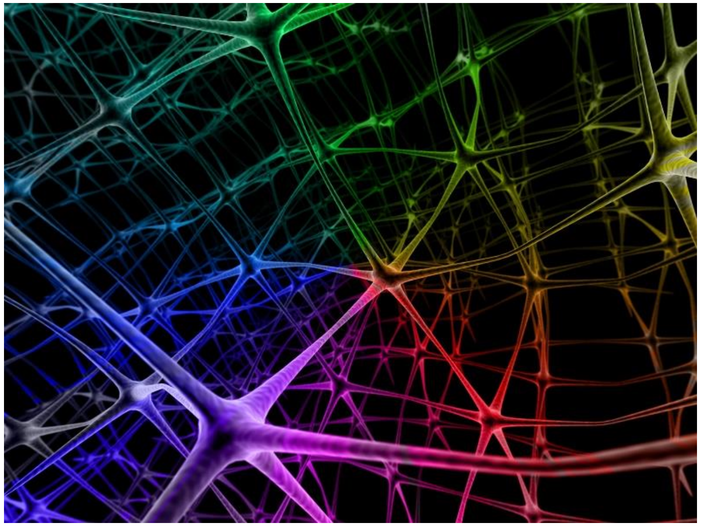
roundworm

brown root
rot fungus

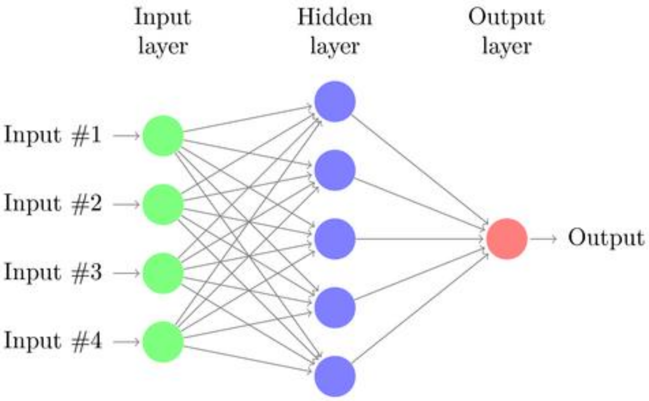
6%

94%

Neural Networks

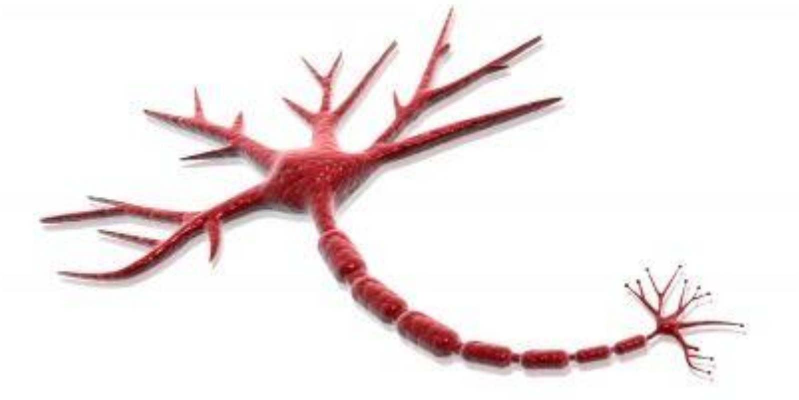


In nature

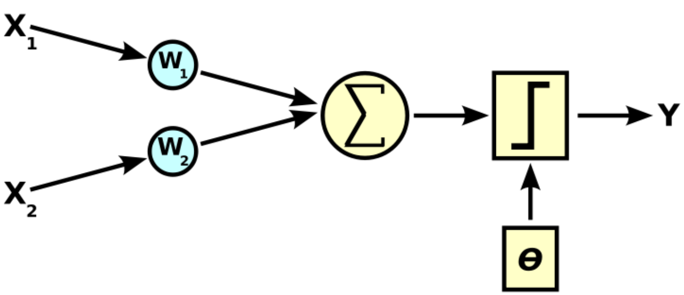


In machine

Neural Networks



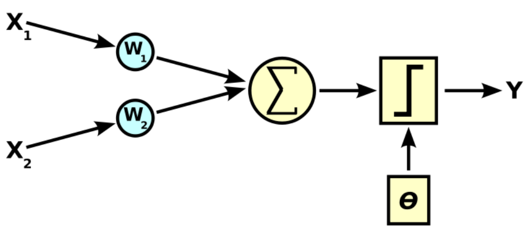
$$f(\mathbf{x}) = \sigma(\mathbf{w}^T \cdot \mathbf{x} + \mathbf{b})$$

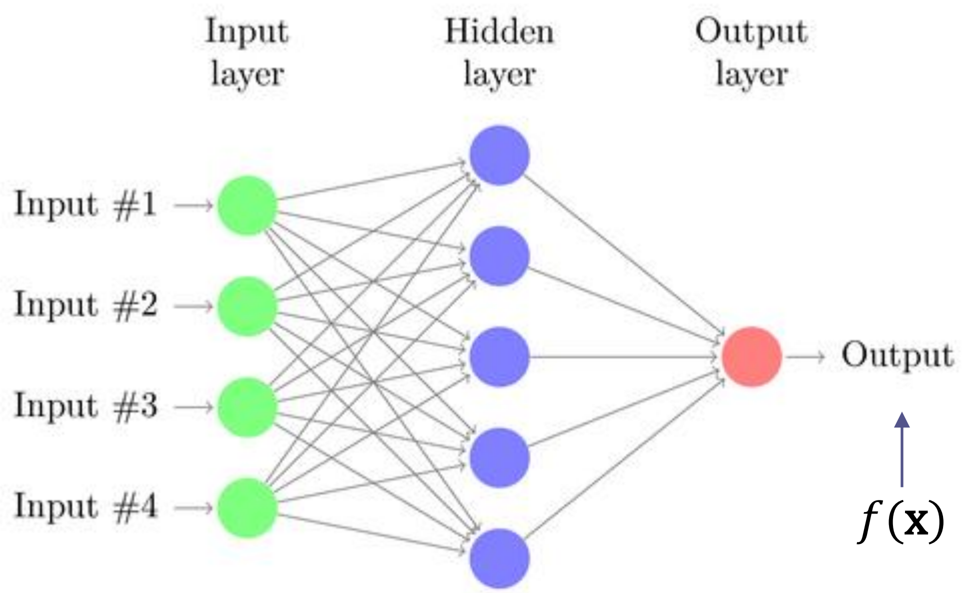


In nature

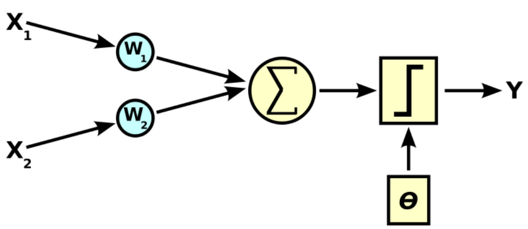
In machine

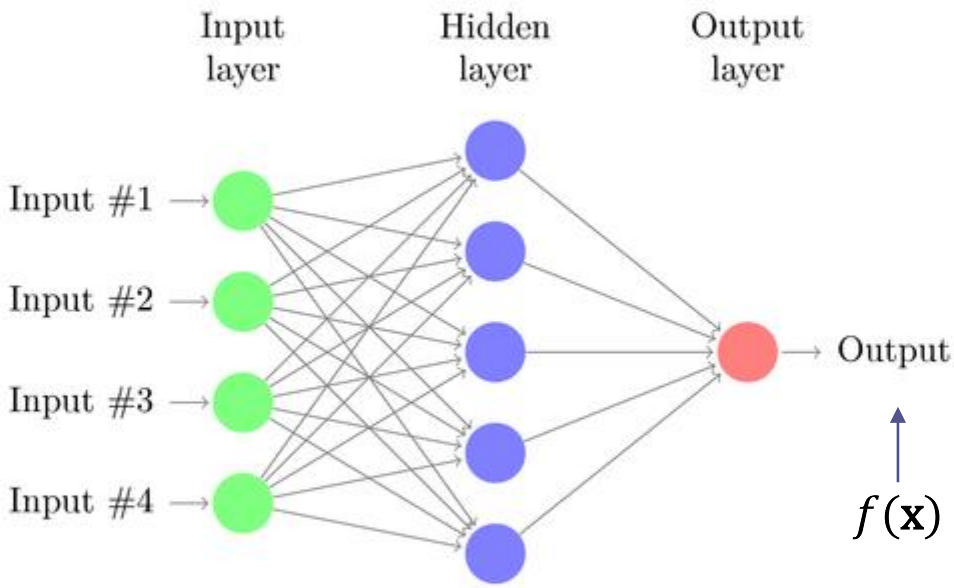
Neural Networks





Neural Networks





How to learn parameters?

Learning Neural Networks

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

1. Use proper activation function σ
   → make f differentiable
2. Define cost function (based on net output)
3. Use backpropagation to compute gradient
   i.e. partial derivatives of cost function w.r.t.
   net parameters
4. Perform gradient descent to find local optima

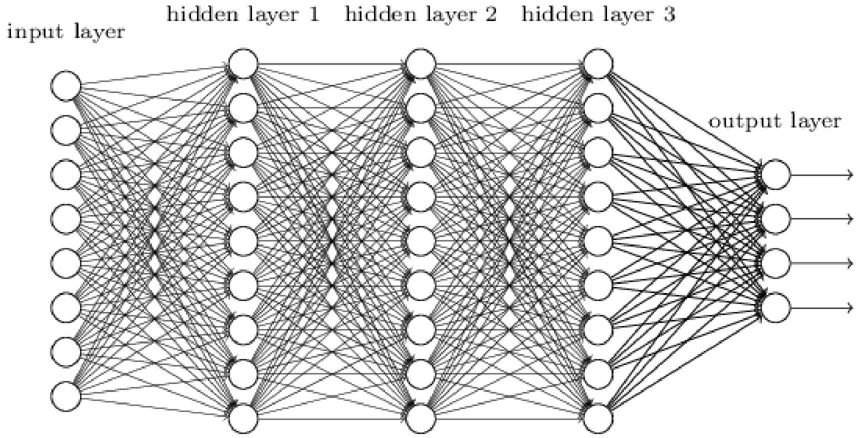
Stochastic Gradient Descent (SGD)

- "Normal" gradient descent
  - Compute gradient using all training examples
    → computationally hard
- Stochastic gradient descent
  - Split the set in mini-batches
  - Use mini-batches to "estimate" gradient

  → significant speedups

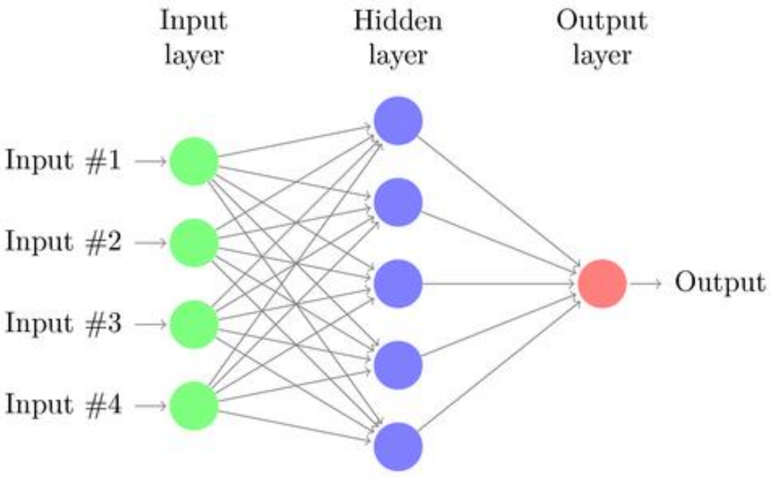
Deep Learning

A class of machine learning techniques that exploit ***many layers*** *of* ***non-linear*** *information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification.*



Universal Approximation Theorem

- Any continuous function can be approximated using a finite feed-forward neural network with a single hidden layer.



Shallow vs Deep Networks

- How to compute parity? Using
  - Shallow network
  - Deep network

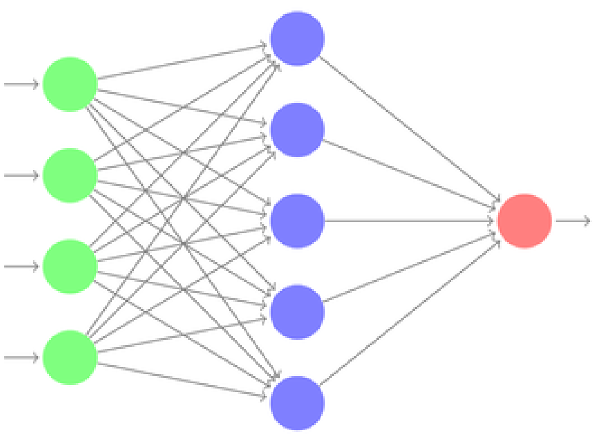
Training Deep Networks is Hard

- Huge number of free parameters
  - Highly prone to overfitting
  - Applies to shallow networks as well

- Unstable gradient
  - Gradient tends to vanish/explode at certain layers
    → different layers learn at different speeds

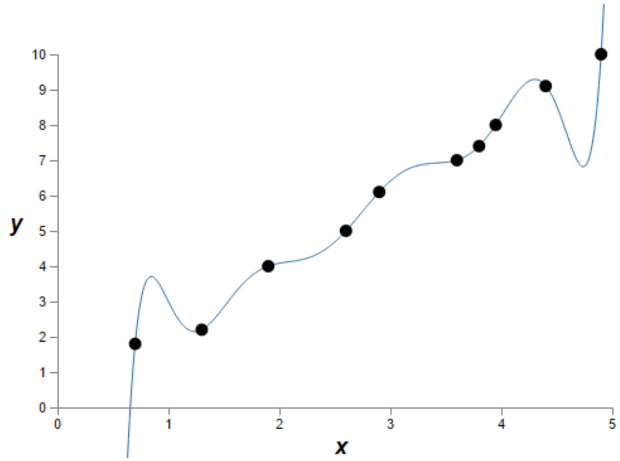
Overfitting Issue

- Fermi: "I remember my friend Johnny von Neumann used to say, with four parameters I can fit an elephant, and with five I can make him wiggle his trunk.“

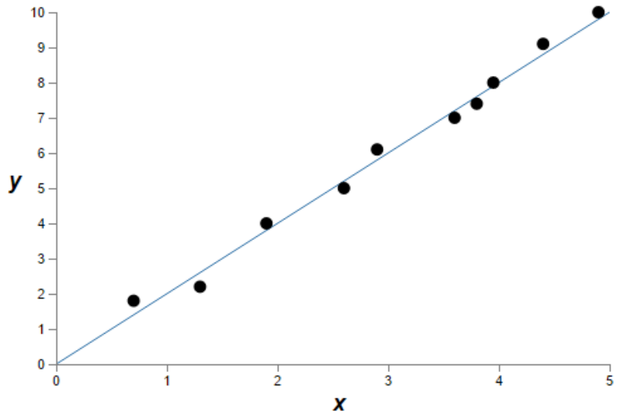


- We need to generalize well, not just fit examples.

Overfitting Issue



vs.



Regularization

- L2-regularization:

$$C = \frac{1}{2n}\sum_x \|y - a^L\|^2$$

$$\Downarrow$$

$$C = \frac{1}{2n}\sum_x \|y - a^L\|^2 + \frac{\lambda}{2n}\|w\|^2$$

Regularization

- Dropout
  - in each step of SGD, force some neurons to zero (selected randomly)

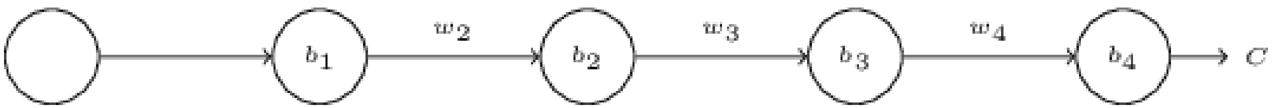
→ similar effect as L2-regularization

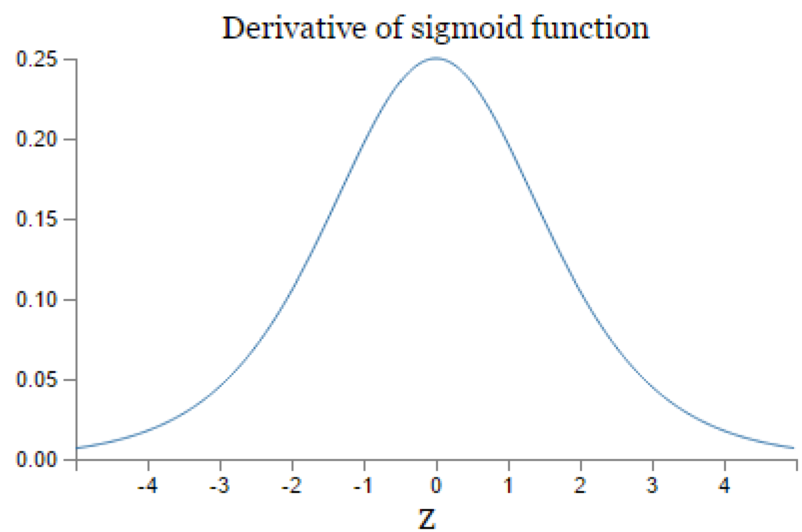
Fabricate Examples

- The more examples presented, the less likely to overfit.

Unstable Gradient

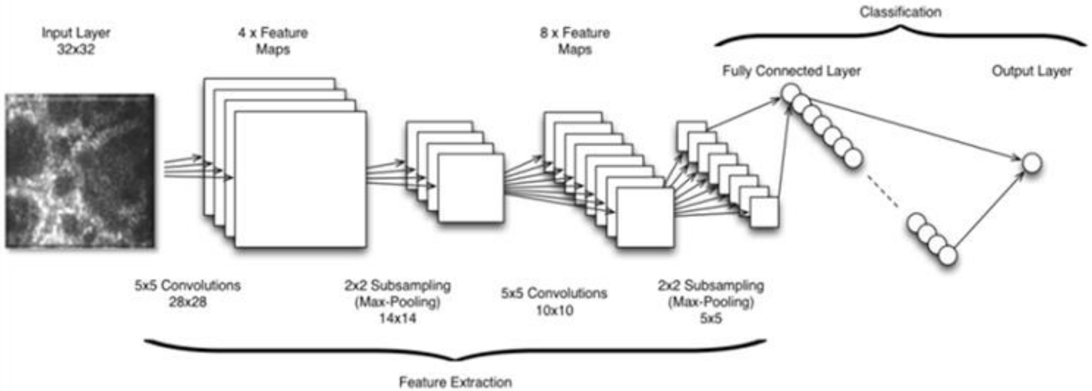
$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$



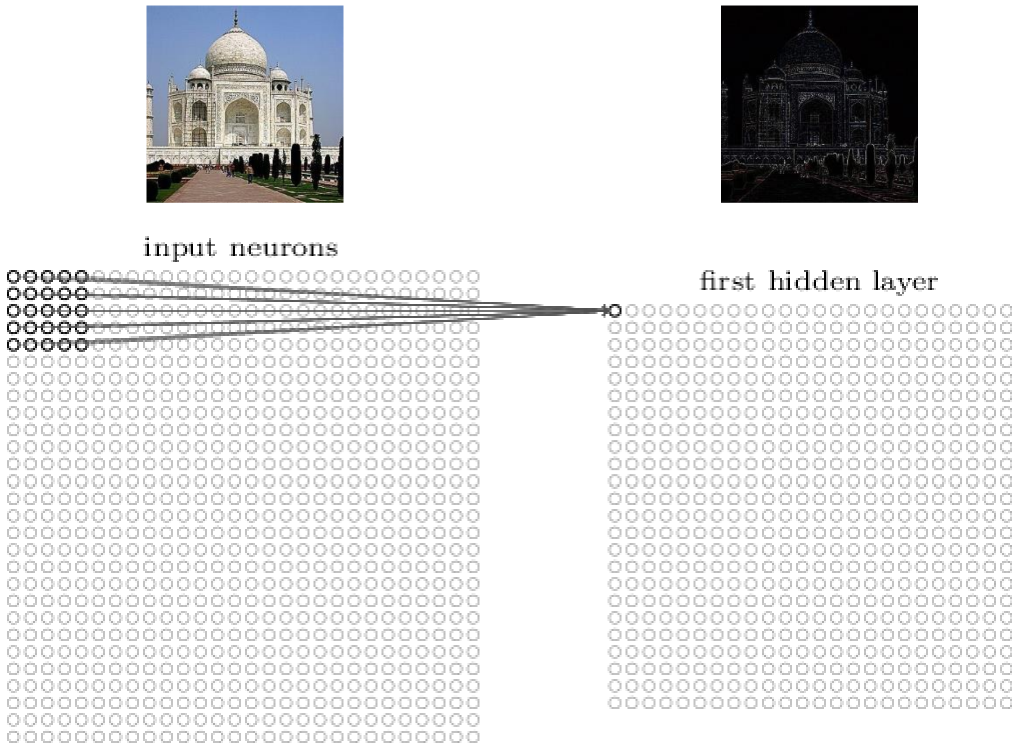


Convolutional Neural Network

- Brief idea:
  - Focus on features on the **local** scale
    → local receptive fields decrease number of parameters
  - Use shared weights
    → "ignore" translations and operate the "same"



Convolutional Layer



Convolutional Layer

• Convolution:

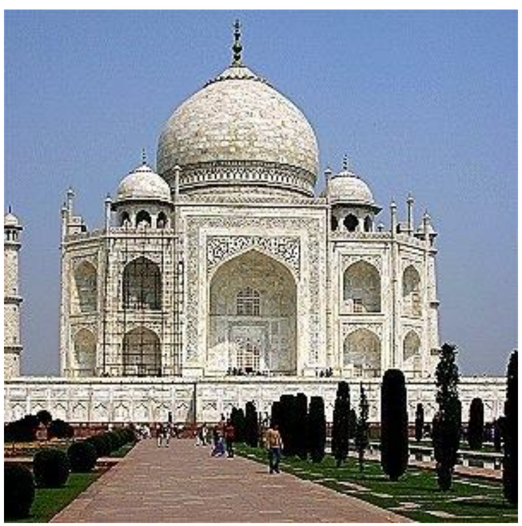
35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

×

	0	1	0	
	0	0	0	
	0	0	0	

=

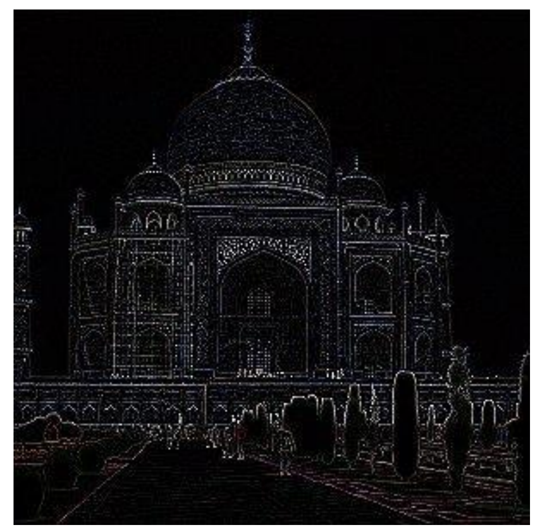
		42		



×

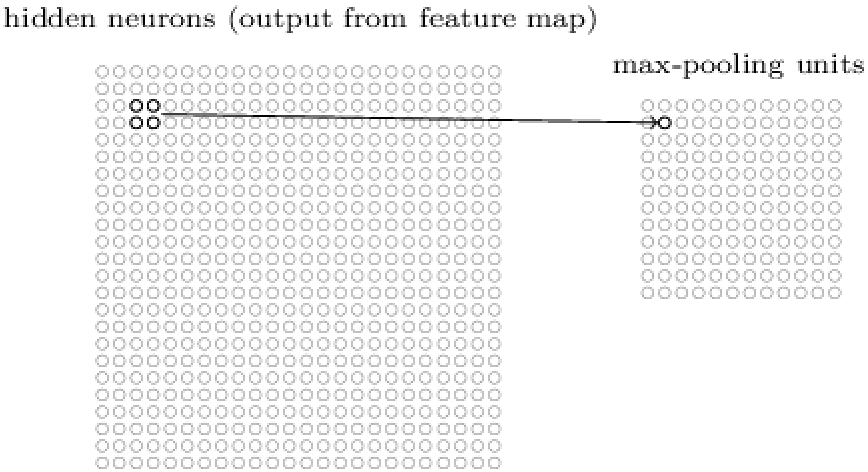
	0	1	0	
	1	-4	1	
	0	1	0	

=



Pooling Layer

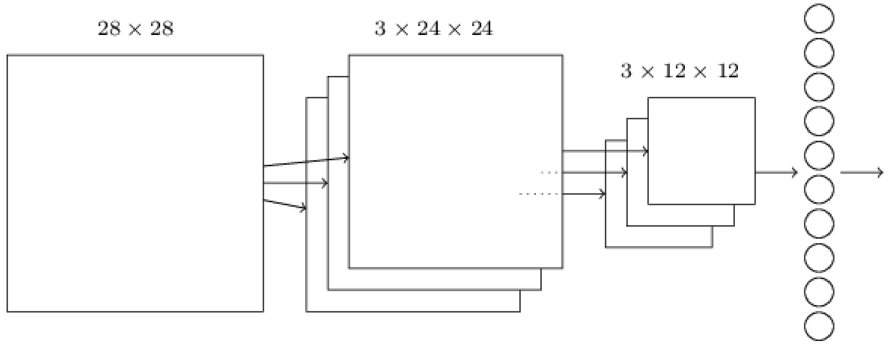
- Condense information

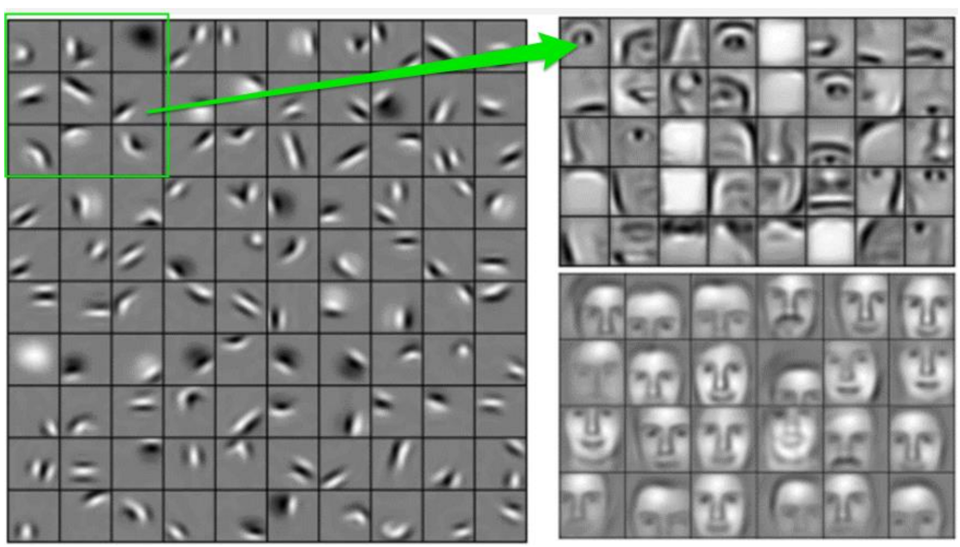


Classification Layer

- “Normal” neurons

Convolutional Neural Network





Convolutional Neural Nework

- How do we avoid problems in learning?
  - Overfitting:
    Shared weights drastically decrease number of free parameters.
  - Unstable gradient:
    Who knows? But it works quite well…

Sequence to Sequence Learning

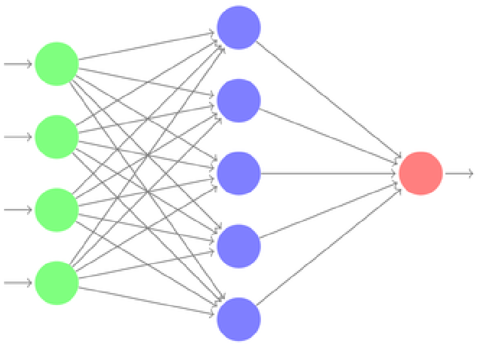
→Neural machine translation

- Training set:

Situace je kritická.	The situation is critical.
Skutečně existují.	They do exist.
Usilujeme-li o dlouhodobou udržitelnost, musíme se zabývat změnou klimatu, a to v oblasti právních předpisů i zdravé ochrany přírodních stanovišť.	If this is to be sustainable in the long term, work on climate change is needed, both legislation and sound protection of habitats.

Sequence to Sequence Learning

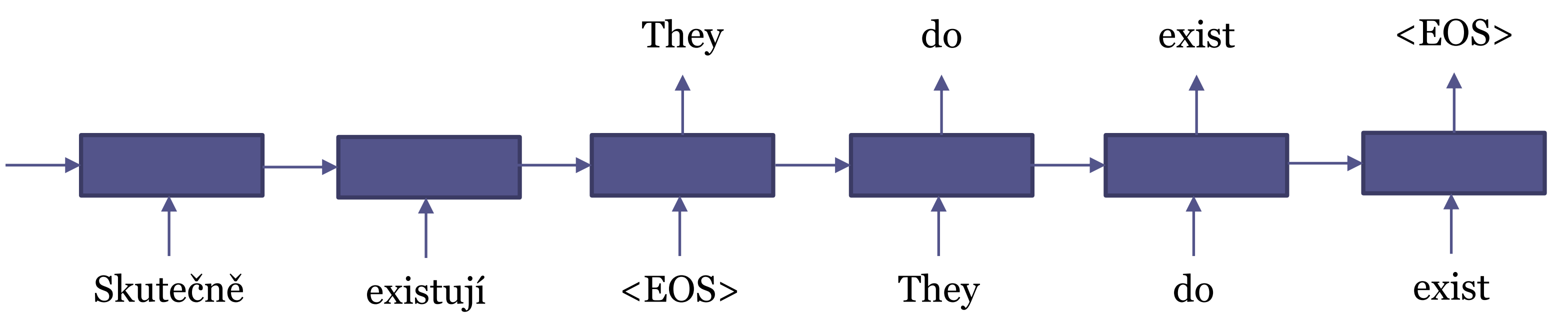
→Neural machine translation



- Training set:

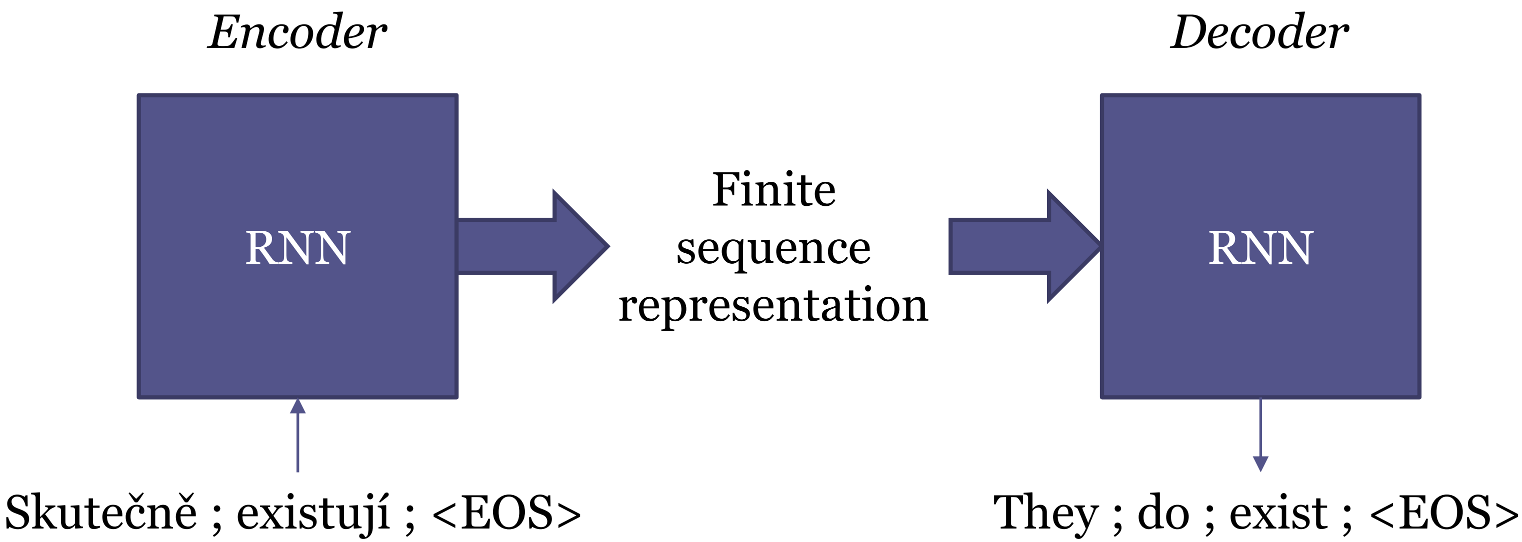
Situace je kritická.	The situation is critical.
Skutečně existují.	They do exist.
Usilujeme-li o dlouhodobou udržitelnost, musíme se zabývat změnou klimatu, a to v oblasti právních předpisů i zdravé ochrany přírodních stanovišť.	If this is to be sustainable in the long term, work on climate change is needed, both legislation and sound protection of habitats.

Sequence to Sequence Learning



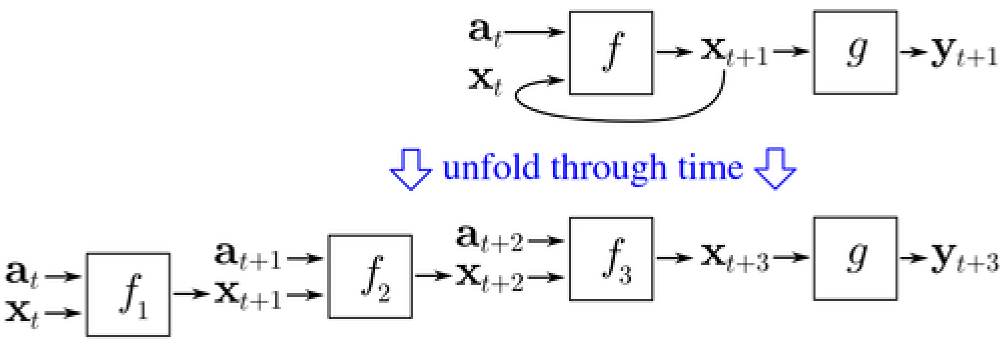
Encoder-Decoder

- Recurrent Neural Network (RNN)
  - activation: $a_t = \sigma(x, a_{t-1})$
  - output: $y_t = \phi(a_t)$



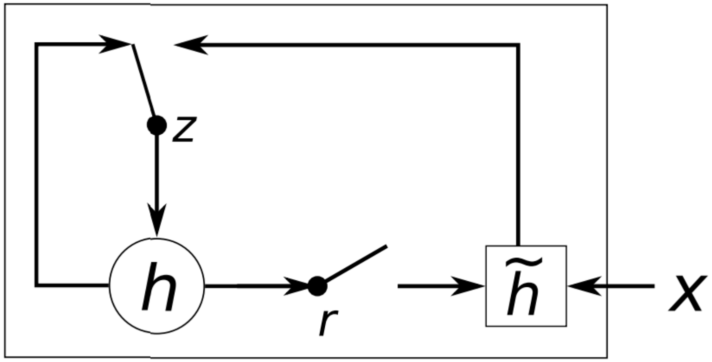
Learning RNNs

- Backpropagation Through Time (BPTT)



  - Main issue: Very deep for long sequences
    → Vanishing/Exploding gradient

Workaround



- Gated "memory" neurons
  - They keep their value unless instructed otherwise
  - If they are wrong, they err for a longer time
    → error does not easily vanish
  - e.g. Long Short-Term Memory cells (LSTM)

Deep LSTM for Machine Translation

Type	Sentence
Our model	Ulrich UNK , membre du conseil d' administration du constructeur automobile Audi , affirme qu' il s' agit d' une pratique courante depuis des années pour que les téléphones portables puissent être collectés avant les réunions du conseil d' administration afin qu' ils ne soient pas utilisés comme appareils d' écoute à distance .
Truth	Ulrich Hackenberg , membre du conseil d' administration du constructeur automobile Audi , déclare que la collecte des téléphones portables avant les réunions du conseil , afin qu' ils ne puissent pas être utilisés comme appareils d' écoute à distance , est une pratique courante depuis des années .
Our model	" Les téléphones cellulaires , qui sont vraiment une question , non seulement parce qu' ils pourraient potentiellement causer des interférences avec les appareils de navigation , mais nous savons , selon la FCC , qu' ils pourraient interférer avec les tours de téléphone cellulaire lorsqu' ils sont dans l' air " , dit UNK .
Truth	" Les téléphones portables sont véritablement un problème , non seulement parce qu' ils pourraient éventuellement créer des interférences avec les instruments de navigation , mais parce que nous savons , d' après la FCC , qu' ils pourraient perturber les antennes-relais de téléphonie mobile s' ils sont utilisés à bord " , a déclaré Rosenker .
Our model	Avec la crémation , il y a un " sentiment de violence contre le corps d' un être cher " , qui sera " réduit à une pile de cendres " en très peu de temps au lieu d' un processus de décomposition " qui accompagnera les étapes du deuil " .
Truth	Il y a , avec la crémation , " une violence faite au corps aimé " , qui va être " réduit à un tas de cendres " en très peu de temps , et non après un processus de décomposition , qui " accompagnerait les phases du deuil " .

Table 3: A few examples of long translations produced by the LSTM alongside the ground truth translations. The reader can verify that the translations are sensible using Google translate.

Deep LSTM for Machine Translation

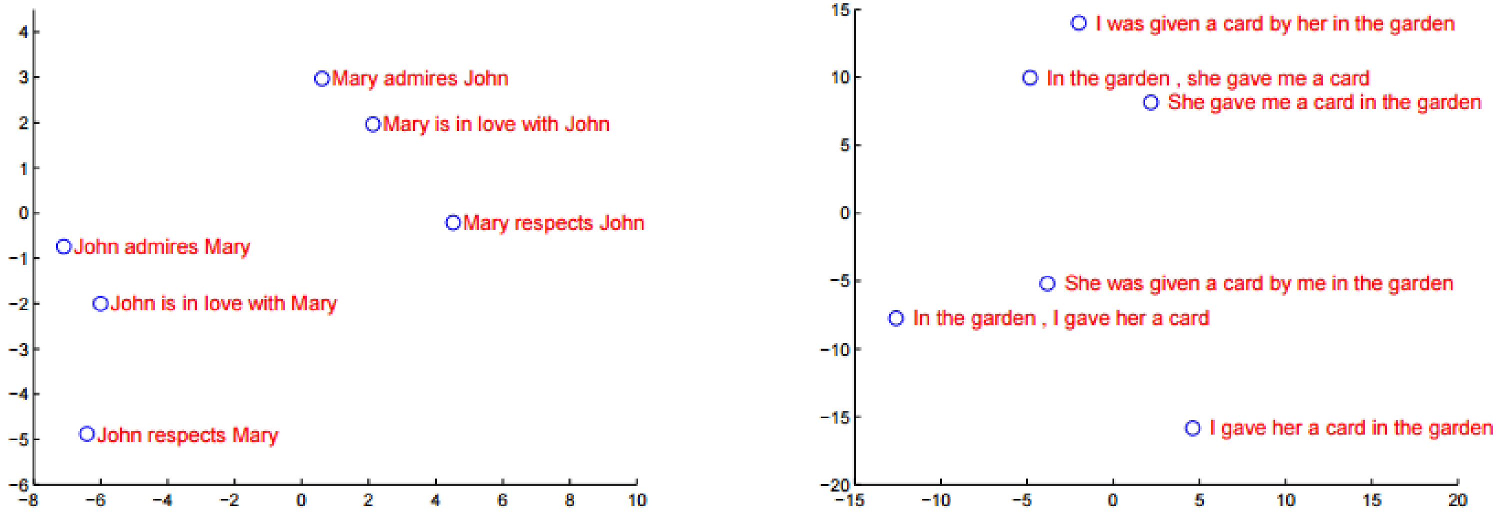
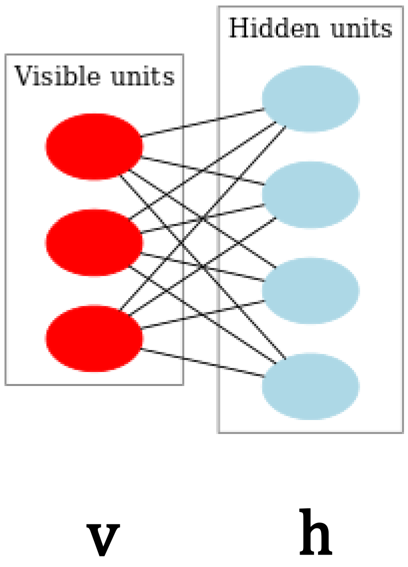


Figure 2: The figure shows a 2-dimensional PCA projection of the LSTM hidden states that are obtained after processing the phrases in the figures. The phrases are clustered by meaning, which in these examples is primarily a function of word order, which would be difficult to capture with a bag-of-words model. Notice that both clusters have similar internal structure.

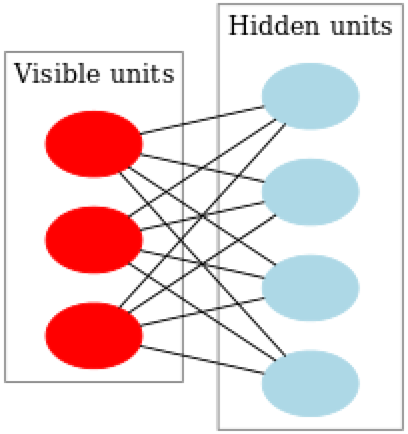
Restricted Boltzmann Machine

- Undirected graphical model
  - Goal: Learn joint probability $P(\mathbf{v}, \mathbf{h})$
  - Tricky part: We do not know **h**
    - Unsupervised learning
    - Find **h** to make the model likely



Restricted Boltzmann Machine

- Each $(\mathbf{v}, \mathbf{h})$ pair has energy:
$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{b}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} - \mathbf{v}^T W\mathbf{h}$$
should be minimal



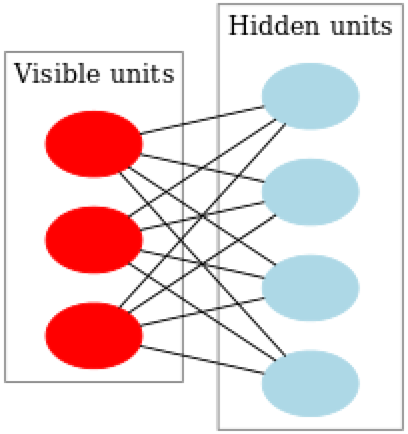
- Probability
$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})}$$
$$P(h|v) = \prod_{j=1}^{n} P(h_j|v) = \prod_{j=1}^{n} \sigma(c_j + \mathbf{v}^T W_j)$$

Restricted Boltzmann Machine

- Objective: maximize log-likelihood

$$\ell(W, \mathbf{b}, \mathbf{c}) = \log \prod_{i=1}^{n} P(\mathbf{v}^t)$$

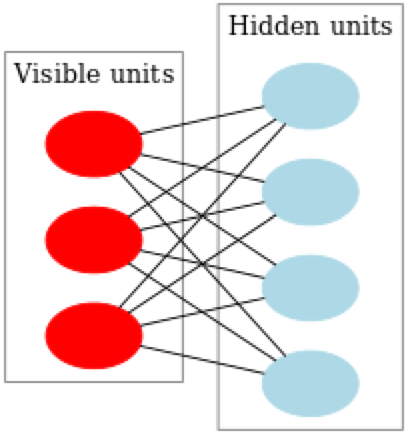
- Gradient ascent optimization



$\mathbf{v}$ $\mathbf{h}$

Restricted Boltzmann Machine

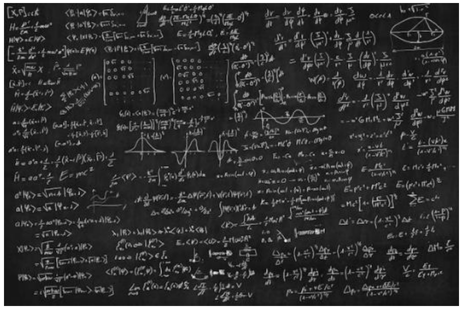
- Usages:
  - Dimensionality reduction
  - Automatic generation of features
  - Reconstruction of incomplete data
    → e.g. collaborative filtering



Deep Belief Network

- RBMs stacked on top of each other
- One way to form a deep autoencoder

Deep Learning

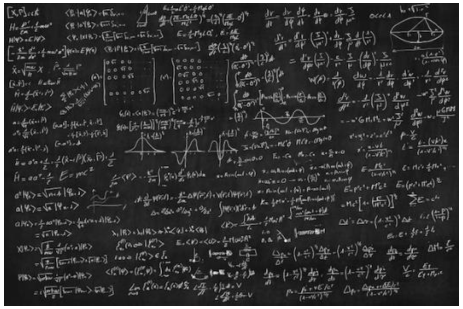


+

Math

Magic

Deep Learning



+

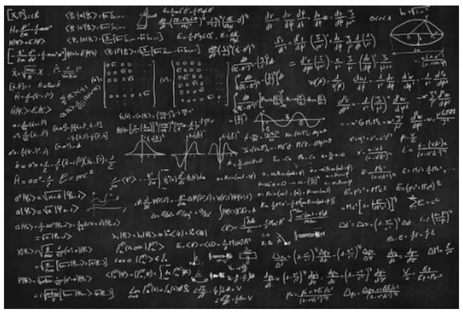
Math

Magic

How to set hyperparameters?

- Network structure
- Activation functions
- Learning rate
- Cost function
- Weight initialization
- …….

Deep Learning



Math

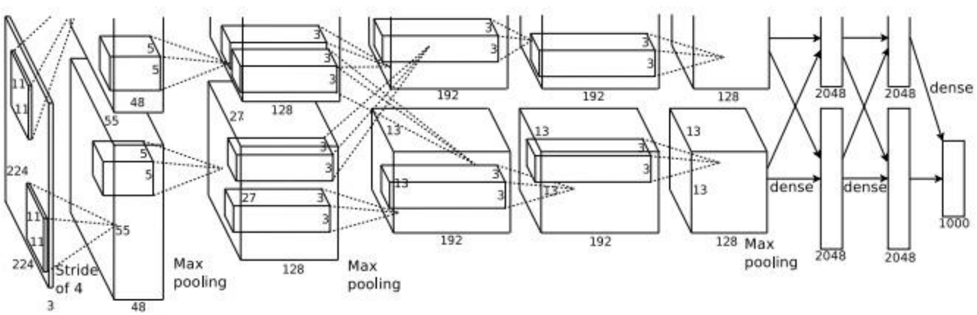
+

Magic

+

HACKS!

How to make the learning efficient?
→ massive parallelization!



Thank you!