Fourth Industrial Summer School

Advanced Machine Learning

Neural Networks and Deep learning-Part3

Session Objectives

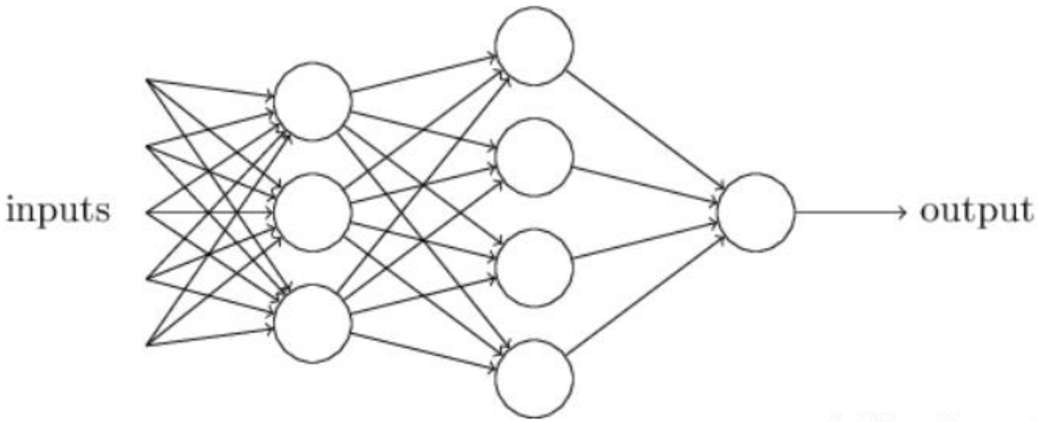
- ✓ Introduction
- ✓ Fundamentals
- ✓ Neural Network Intuitions
- ✓ 2-Layer Neural Network
- ✓ Deep Neural Networks

Deep Neural Networks

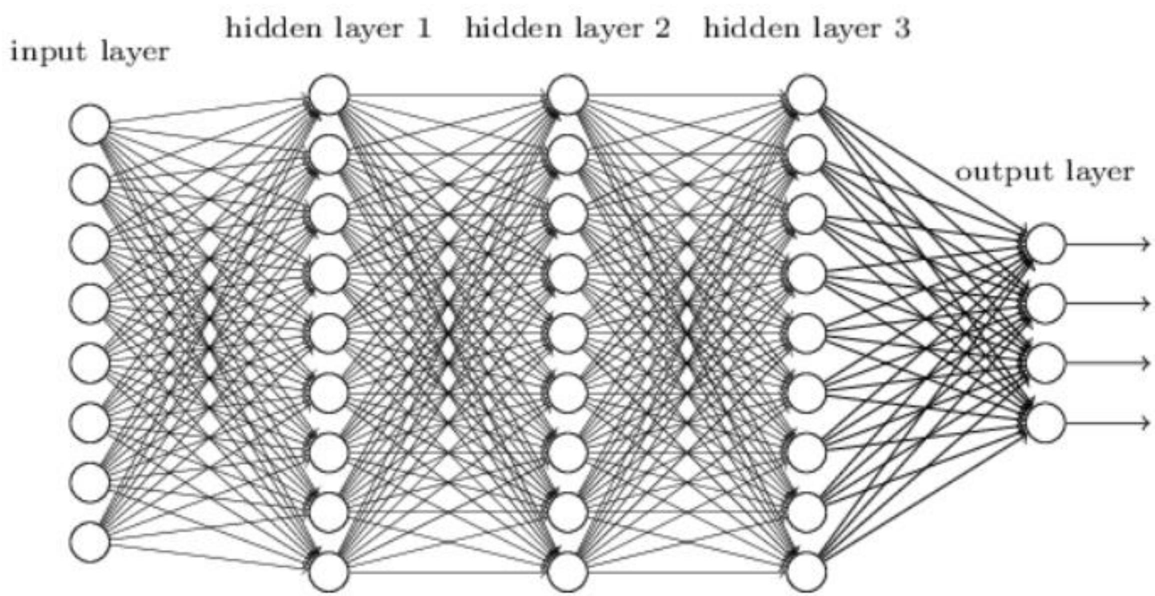
Basic Architecture

What is a deep neural network

- With more than one hidden layer. E.g.:



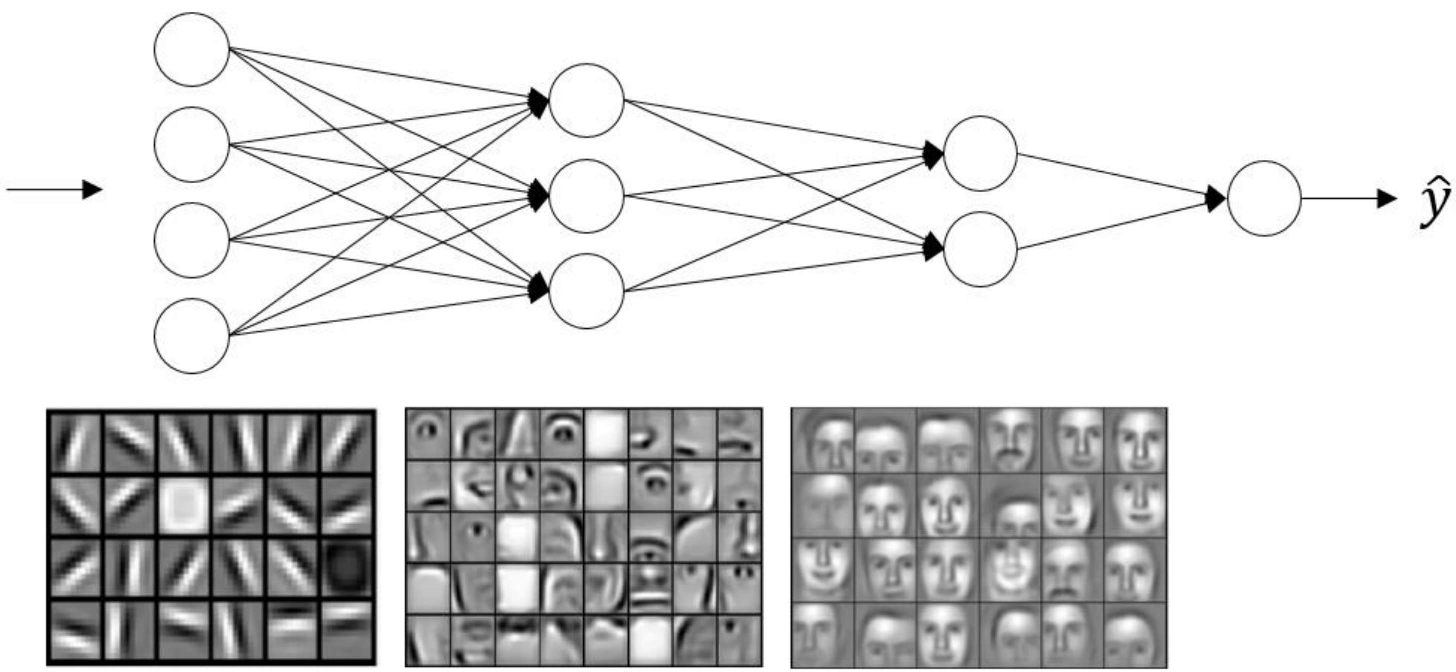
2-layer fully connected



3-layer fully connected

Why deep neural networks

- They seem to learn complex structures in stages starting from simple features to complex features
- Think about autoencoder intuition



Computations involved

- Forward propagations
- Cost computations
- Backward propagations

Hyper-parameters

- Learning rate
- Number of layers
- Neuron/layers
- Activation function
- Others like number of iterations, optimization strategy.

Bias

- A more complex model: More neurons, layers.
- Train longer
- More features

Variance

- More data
- L_p regularization
- Dropout
- Early stopping
- Data augmentation

Practical issues

- Scaling
- Vanishing/exploding gradients
  - Weight initialization:
  - weights = np.random.randn(dim) * np.sqrt(2/n)), or:
  - weights = np.random.randn(dim) * np.sqrt(1/n)), [Xavier initialization]
- Batch/mini-batch/stochastic gradient descent

Optimization algorithms- Gradient descent with momentum

$$v_{dW} = \beta v_{dW} + (1-\beta)dW$$

$$v_{db} = \beta v_{db} + (1-\beta)db$$

$$W = W - \alpha v_{dW},$$

$$b = b - \alpha v_{db}$$

- β is normally set to 0.9
- Bias correction

RMSprop

$$s_{dW} = \beta s_{dW} + (1 - \beta) dW^{\,2}$$

$$s_{db} = \beta s_{db} + (1 - \beta) db^{\,2}$$

$$W = W - \alpha \frac{dw}{\sqrt{s_{dW}} + \epsilon},$$

$$b = b - \alpha \frac{db}{\sqrt{s_{db}} + \epsilon}$$

Adam (Adaptive moment estimation)

$$v_{dW} = \beta_1 v_{dW} + (1 - \beta_1) dW$$

$$v_{db} = \beta_1 v_{db} + (1 - \beta_1) db$$

$$s_{dW} = \beta_2 s_{dW} + (1 - \beta_2) dW^{\,2}$$

$$s_{db} = \beta_2 s_{db} + (1 - \beta_2) db^{\,2}$$

$$v_{dw}^{corr} = \frac{v_{dW}}{(1-\beta_1^t)}$$

$$v_{db}^{corr} = \frac{v_{db}}{(1-\beta_1^t)}$$

$$s_{dw}^{corr} = \frac{s_{dW}}{(1-\beta_2^t)}$$

$$s_{db}^{corr} = \frac{s_{db}}{(1-\beta_2^t)}$$

$$W = W - \alpha \frac{v_{dw}^{corr}}{\sqrt{s_{dw}^{corr}} + \epsilon},$$

$$b = b - \alpha \frac{v_{db}^{corr}}{\sqrt{s_{db}^{corr}} + \epsilon}$$

Animation: http://ruder.io/optimizing-gradient-descent/
Interactive: https://damip.net/article-gradient-descent-comparison

Learning rate decay

$$\alpha = \frac{1}{1 + decay\_rate * epoc\_num} \alpha_0$$

Or

$$\alpha = k^{epoc\_num} * \alpha_0$$

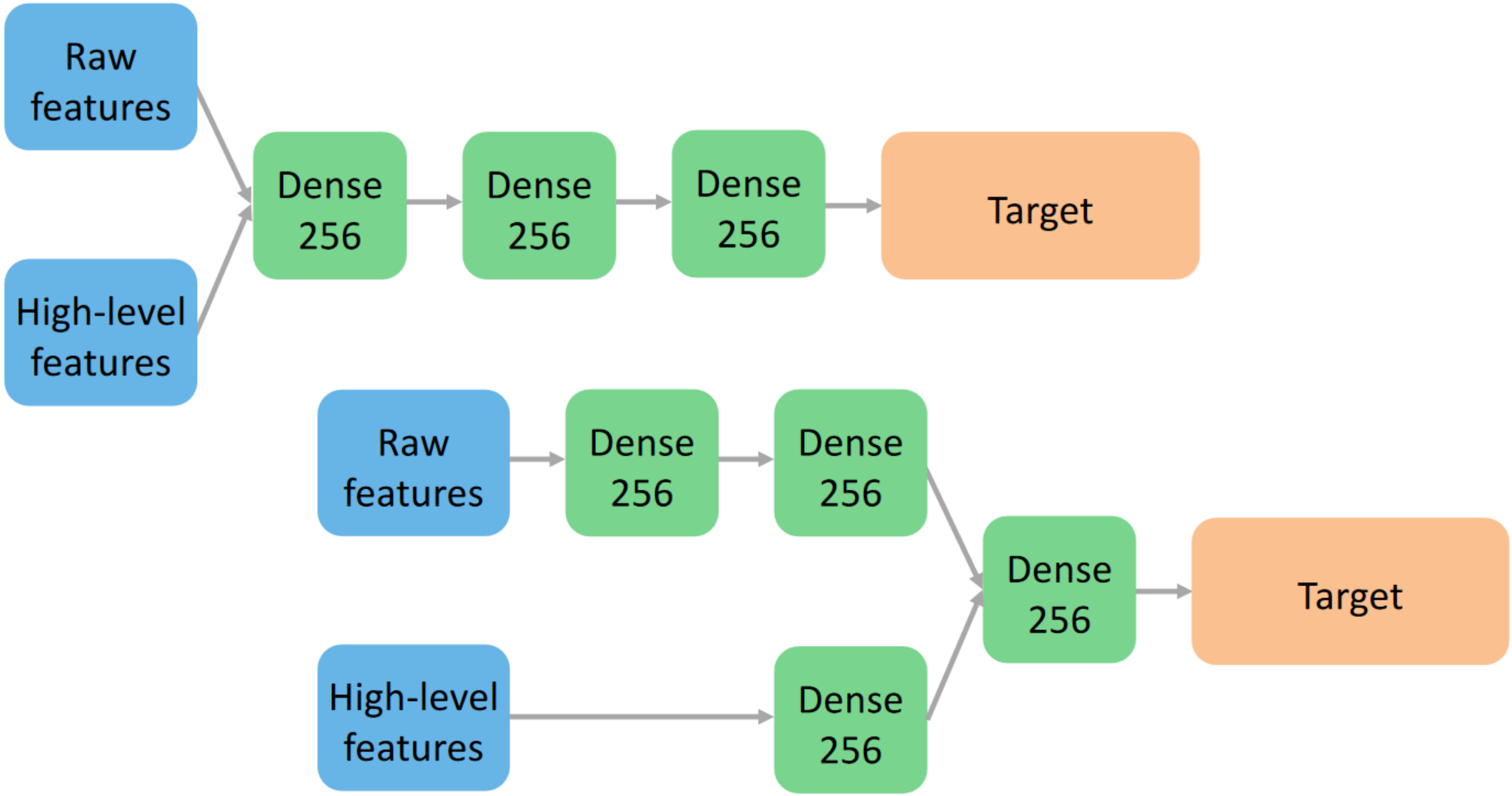
Batch normalization

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \text{, and } \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2.$$

$$\hat{x}_i^{(k)} = \frac{x_i^{(k)} - \mu_B^{(k)}}{\sqrt{\sigma_B^{(k)^2} + \epsilon}},$$

$$y_i^{(k)} = \gamma^{(k)}\hat{x}_i^{(k)} + \beta^{(k)},$$

Deep learning is a language



What if I want to give less weight to images?

Exercise

- https://playground.tensorflow.org

References

- Andrew Ng, Neural Networks and Deep Learning, Stanford University
- Introduction to Deep Learning, National Research University Higher School of Economics
- Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015
- https://playground.tensorflow.org
- Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar, Foundations of Machine Learning, second edition, The MIT Press
- Andrew Ng, Machine Learning Yearning, deeplearning.ai