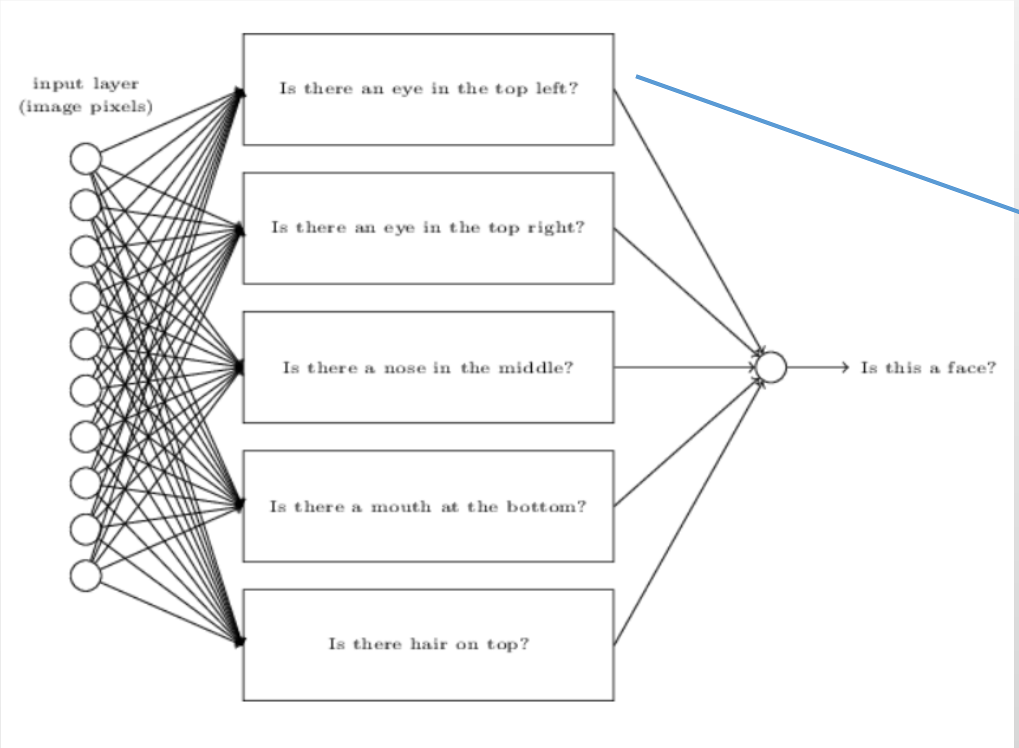
Presentation about Deep Learning

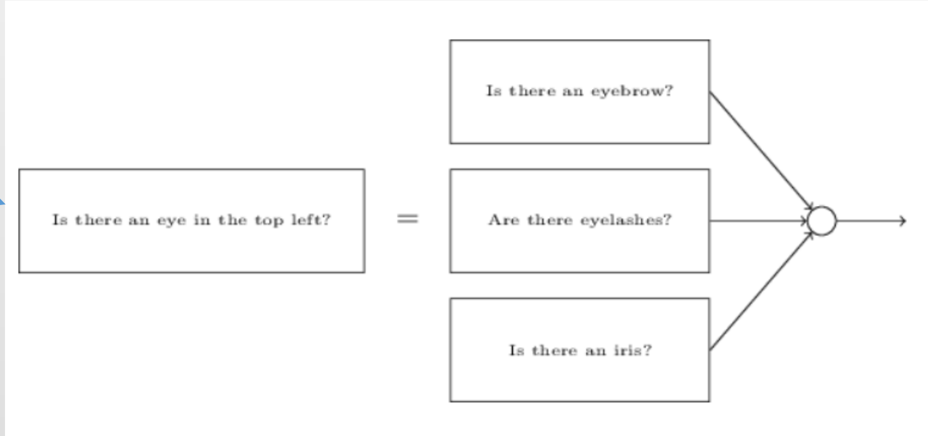
--- Zhongwu xie

Contents

Deep learning

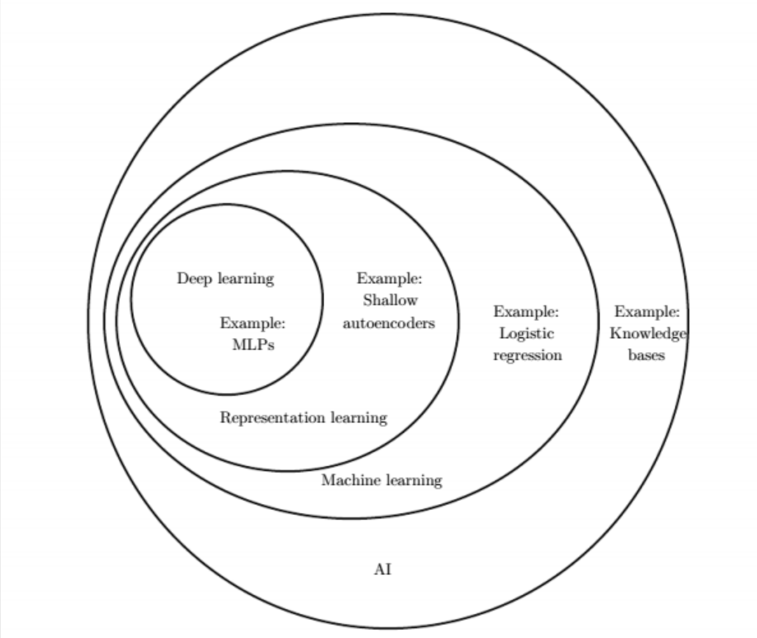
I . Introduction to Deep Learning

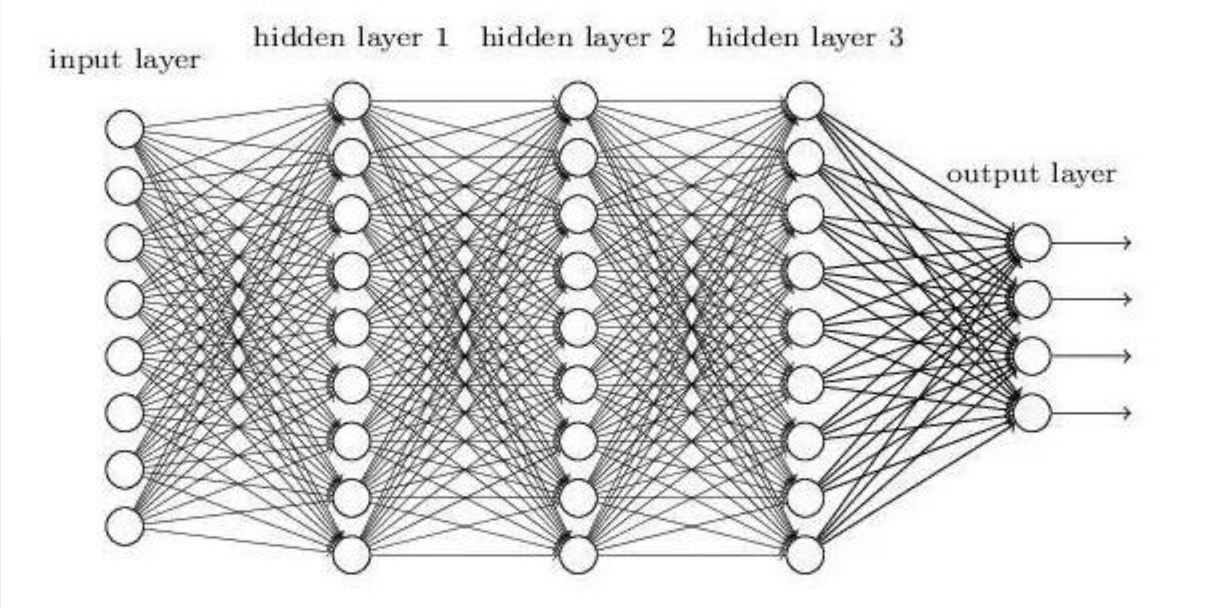




Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts , with each concept defined in relation to simpler concepts , and more abstract representations computed in terms of less abstract ones.---Ian Goodfellow

I . Introduction to Deep Learning





In the plot on the left , A Venn diagram showing how deep learning is a kind of representation learning , which is in turn of machine learning. In the plot on the left ,the graph shows that deep learning has Multilayer.

I . What is Deep Learning

- Data: $(x_i, y_i)\quad 1 \le i \le m$
- Model: ANN
- Criterion:

  -Cost function: $L(y, f(x))$

  -Empirical risk minimization: $R(\theta) = \frac{1}{m}\sum_{i=1}^{m} L(y_i, f(x_i, \theta))$

  -Regularization: $||w||$,$||w||^2$, Early Stopping , Dropout

  -objective function: $mini\ R(\theta) + \lambda *$(Regularization Function)
- Algorithm : BP Gradient descent

Learning is cast as optimization.

II . Why should we need to learn Deep Learning? --- Efficiency

- Speech Recognition

famous Instances : self-driven
AlphaGo

---The phoneme error rate on TIMIT:

Basing on HMM-GMM in 1990s : about 26%

Restricted Boltzmann machines(RBMs) in 2009: 20.7%; LSTM-RNN in 2013:17.7%

- Computer Vision

---The Top-5 error of ILSVRC 2017 Classification Task is 2.251%, while human being' s is 5.1%.

- Natural Language Processing

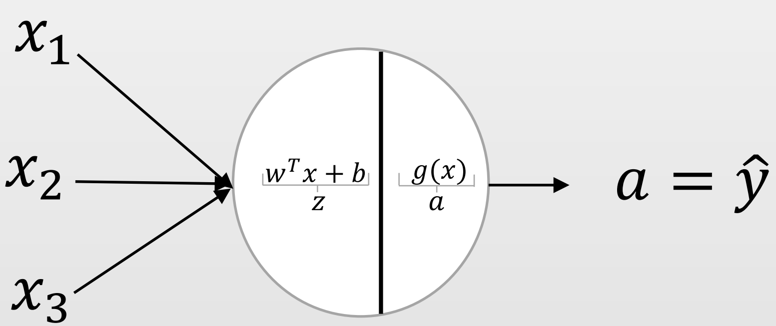
---language model (n-gram) Machine translation

- Recommender Systems

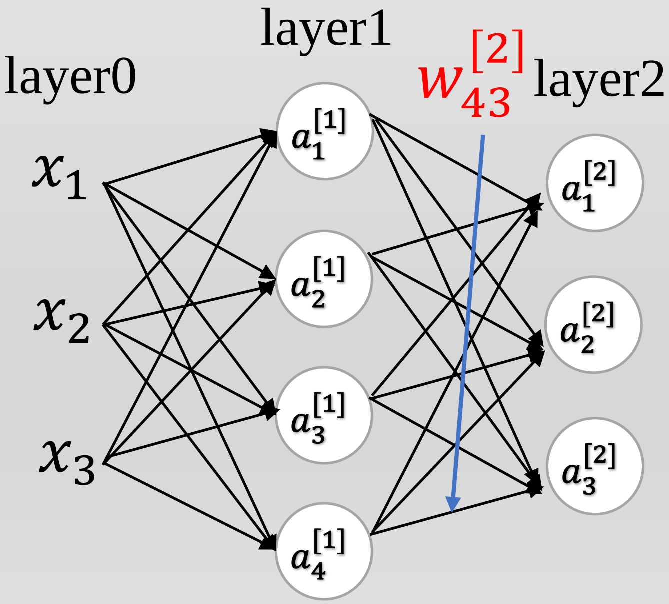
---Recommend ads , social network news feeds , movies , jokes , or advice from experts etc.

Backward propagation

Ⅰ. Introduction to Notation

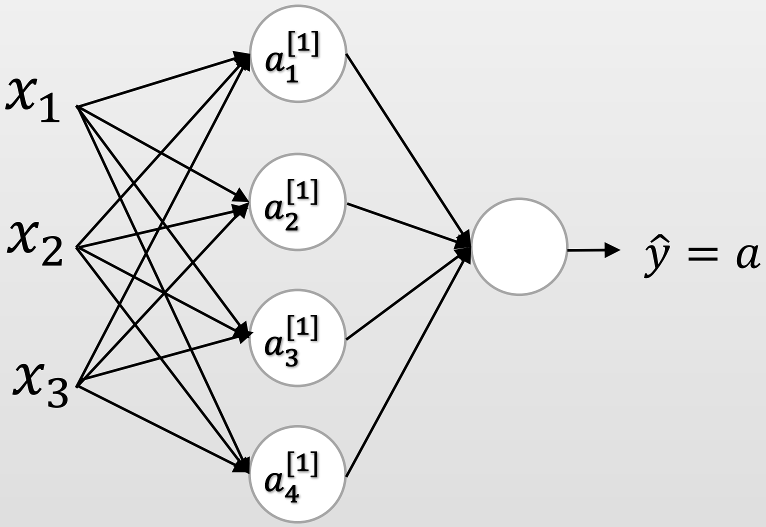


$$z = w^T x + b$$
$$a = g(z)$$



w_{jk}^{l} is the weight from the j^{th} neuron in the $(l-1)^{th}$ layer to the k^{th} neuron in the l^{th} layer.

Ⅰ. Introduction to Forward propagation and Notation



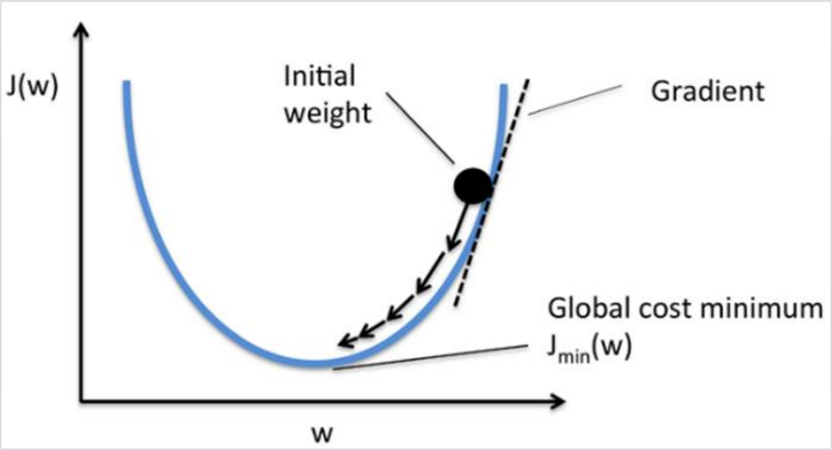
$$z_1^{[1]} = w_1^{[1]T} x + b_2^{[1]}, \qquad a_1^{[1]} = \sigma(z_1^{[1]})$$

$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]}, \qquad a_2^{[1]} = \sigma(z_2^{[1]})$$

$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]}, \qquad a_3^{[1]} = \sigma(z_3^{[1]})$$

$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]}, \qquad a_4^{[1]} = \sigma(z_4^{[1]})$$

$$z^{[1]} = \overbrace{\begin{bmatrix} w_{11}^{[1]} & w_{12}^{[1]} & w_{13}^{[1]} & w_{14}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} & w_{23}^{[1]} & w_{24}^{[1]} \\ w_{31}^{[1]} & w_{32}^{[1]} & w_{33}^{[1]} & w_{34}^{[1]} \end{bmatrix}^{T}}^{w^{[1]}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{3} w_{k1}^{[1]} x_k + b_1^{[1]} \\ \sum_{k=1}^{3} w_{k2}^{[1]} x_k + b_2^{[1]} \\ \sum_{k=1}^{3} w_{k3}^{[1]} x_k + b_3^{[1]} \\ \sum_{k=1}^{3} w_{k4}^{[1]} x_k + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix}$$



$$cost\ function: L(a, y)$$

$$dw^{[1]} = \frac{\partial L(a,y)}{\partial w^{[1]}}, \; db^{[1]} = \frac{\partial L(a,y)}{\partial b^{[1]}}$$

II . Backward propagation.

---the chain rule

If $x = f(w), y = f(x), z = f(y)$

So, $\dfrac{\partial z}{\partial w} = \dfrac{\partial z}{\partial y}\dfrac{\partial y}{\partial x}\dfrac{\partial x}{\partial w}$

---the functions of neural network are same as the above function , so we can use the chain rule to the gradient of the neural network.

$$\begin{array}{l} x \searrow \\ w \rightarrow \boxed{z = w^T x + b} \rightarrow \boxed{a = \sigma(z)} \rightarrow \boxed{L(a, y)} \\ b \nearrow \end{array}$$

II . Backward propagation.

---the chain rule

$$L(a,y) = -[yloga + (1-y)\log(1-a)]$$

$$x,\ w^{[1]},\ b^{[1]} \rightarrow \boxed{z^{[1]} = w^{[1]}x + b^{[1]}} \rightarrow \boxed{a^{[1]} = \sigma\left(z^{[1]}\right)} \rightarrow \boxed{z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}} \rightarrow \boxed{a^{[2]} = \sigma\left(z^{[2]}\right)} \rightarrow \boxed{L\left(a^{[2]}, y\right)}$$

$$w^{[2]},\ b^{[2]}$$

$$da^{[2]} = \frac{\partial L(a,y)}{\partial a^{[2]}} = -\frac{y}{a} + \frac{1-y}{1-a}$$

$$dz^{[2]} = \frac{\partial L(a,y)}{\partial z^{[2]}} = \frac{\partial L(a,y)}{\partial a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} = a^{[2]} - y$$

$$dw^{[2]} = \frac{\partial L(a,y)}{\partial w^{[2]}} = \frac{\partial L(a,y)}{a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} \times \frac{\partial z^{[2]}}{\partial w^{[2]}} = dz^{[2]}a^{[1]T}$$

$$db^{[2]} = \frac{\partial L(a,y)}{\partial b^{[2]}} = \frac{\partial L(a,y)}{a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} \times \frac{\partial z^{[2]}}{\partial b^{[2]}} = dz^{[2]}$$

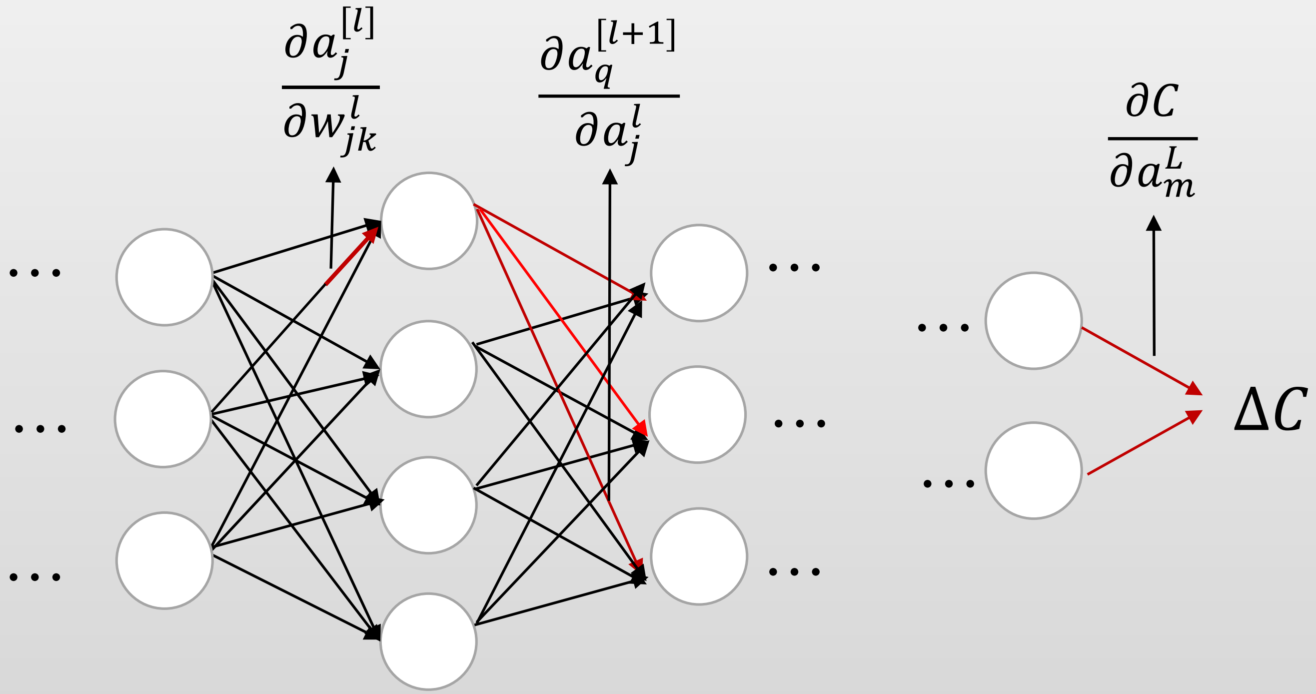
$$dz^{[1]} = \frac{\partial L(a,y)}{\partial z^{[1]}} = \frac{\partial L(a,y)}{a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} \times \frac{\partial z^{[2]}}{\partial a^{[1]}} \times \frac{\partial a^{[1]}}{\partial z^{[1]}}$$

$$= w^{[2]T} dz^{[2]} * \sigma'(z^{[1]})$$

$$dw^{[1]} = \frac{\partial L(a,y)}{\partial w^{[1]}} = \frac{\partial L(a,y)}{\partial a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} \times \frac{\partial z^{[2]}}{\partial a^{[1]}} \times \frac{\partial a^{[1]}}{\partial z^{[1]}} \times \frac{\partial z^{[1]}}{\partial w^{[1]}} = dz^{[1]}x^T$$

$$db^{[1]} = \frac{\partial L(a,y)}{\partial b^{[1]}} = \frac{\partial L(a,y)}{a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} \times \frac{\partial z^{[2]}}{\partial a^{[1]}} \times \frac{\partial a^{[1]}}{\partial z^{[1]}} \times \frac{\partial z^{[1]}}{\partial b^{[1]}} = dz^{[1]}$$

II . Summary : The Backpropagation



$$\Delta C \approx \sum_{mnp..q} \frac{\partial C}{\partial a_m^L}\frac{\partial a_m^L}{\partial a_n^{L-1}}\frac{\partial a_n^{L-1}}{\partial a_p^{L-2}}\cdots\frac{\partial a_q^{l+1}}{\partial a_j^l}\frac{\partial a_j^{[l]}}{\partial w_{jk}^l}\Delta w_{jk}^l$$

$$\frac{\partial C}{\partial w_{jk}^l} = \sum_{mnp..q} \frac{\partial C}{\partial a_m^L}\frac{\partial a_m^L}{\partial a_n^{L-1}}\frac{\partial a_n^{L-1}}{\partial a_p^{L-2}}\cdots\frac{\partial a_q^{l+1}}{\partial a_j^l}\frac{\partial a_j^{[l]}}{\partial w_{jk}^l}$$

The backpropagation algorithm is a clever way of keeping track of small perturbations the weights (and biases) as they propagate through the network , reach the output , and then affect the cost.

---Michael Nielsen

Ⅱ . Summary : The Backpropagation algorithm

1.Input x:Set the corresponding activation for the input layer.

2.Feedforward : For each $l = \mathbf{2, 3, \ldots, L}$ compute $z^{[l]}=w^{[l]}a^{[l-1]} + b^{[l]}$ and $a^{[l]}= \sigma\left(z^{[l]}\right)$.

3.Output error $dz^{[L]}$:$dz^{[L]}=a^{[L]}- y$.

4.Back propagate the cost error:For each l=L-1,L-2,…2 compute : $\mathrm{dz}^{[l]}=(\mathrm{w}^{[l+1]})^{T}\mathrm{dz}^{[l+1]} * \sigma'(\mathrm{z}^{[l]})$

5.Output : The gradient of the cost function is given by：

$dw^{[l]} = \frac{\partial L(a,y)}{\partial w^{[l]}}=dz^{[l]}a^{[l-1]T}$ and $db^{[l]} = \frac{\partial L(a,y)}{\partial b^{[l]}} = dz^{[l]}$

Update the $w_{jk}^{[l]}$and $b_j^{[l]}$：

$$w_{jk}^{[l]}=w_{jk}^{[l]} - \alpha \frac{\partial L(a,y)}{\partial w_{jk}^{[l]}}$$

$$b_j^{[l]}= b_j^{[l]}-\alpha \frac{\partial L(a,y)}{\partial \, b_j^{[l]}}$$

Convolutional Neural Networks

1 . Types of layers in a convolutional network.

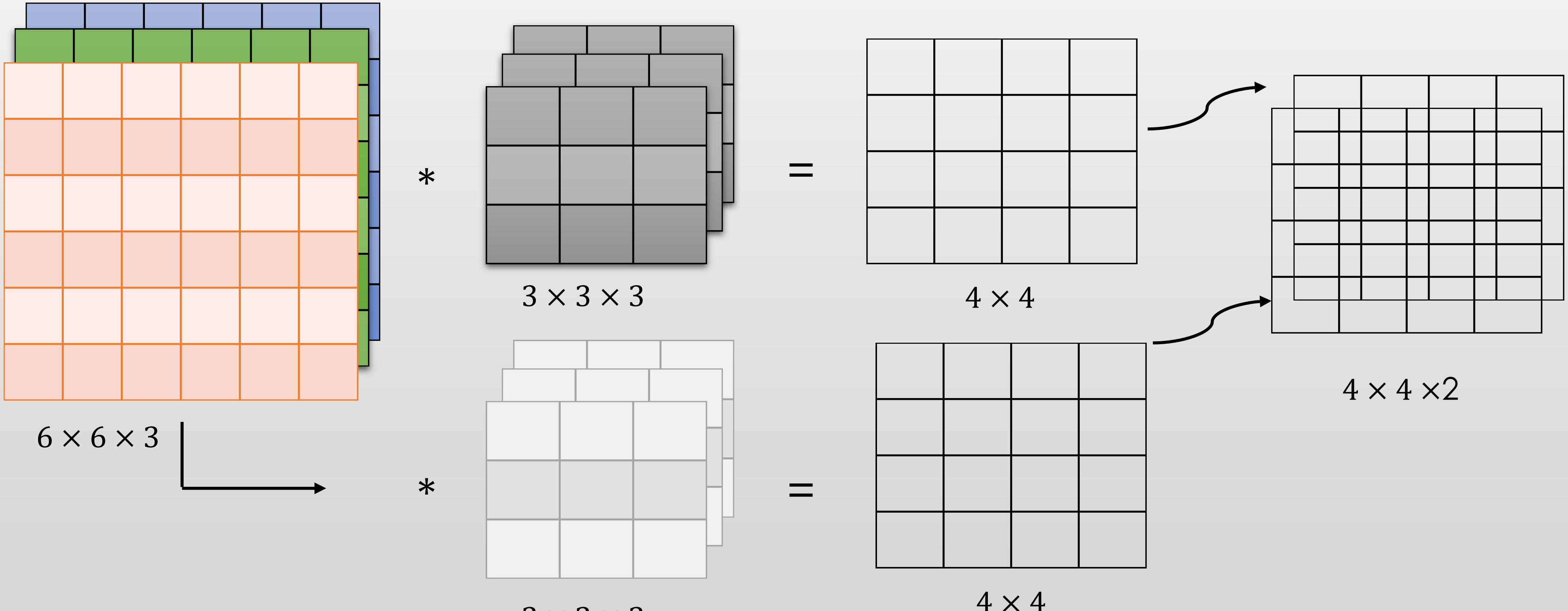
- -Convolution
- -Pooling
- -Fully connected

2.1 Convolution in Neural Network

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \boxed{0}$$

2.2 Multiple filters



Why convolutions?

---Parameter sharing

---Sparsity of connections

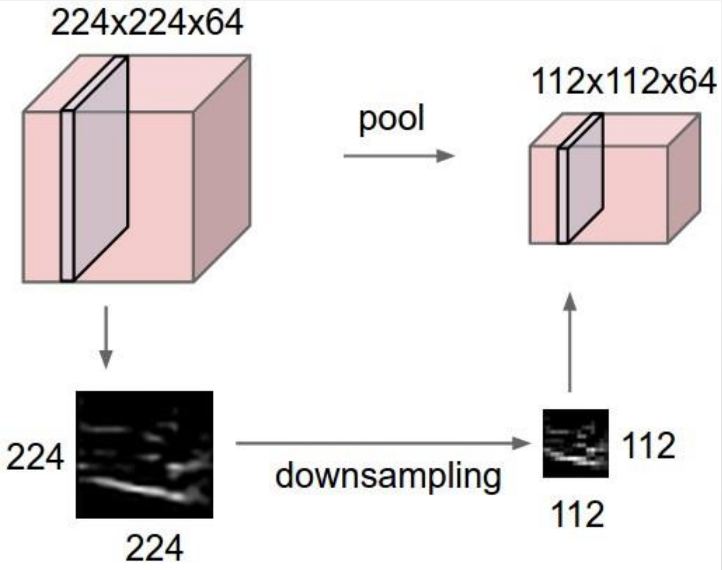
3 . Pooling layers

- Max pooling

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

Max pool with 2 ×2 filters and stride 2 →

9	2
6	3



- Remove the redundancy information of convolutional layer .

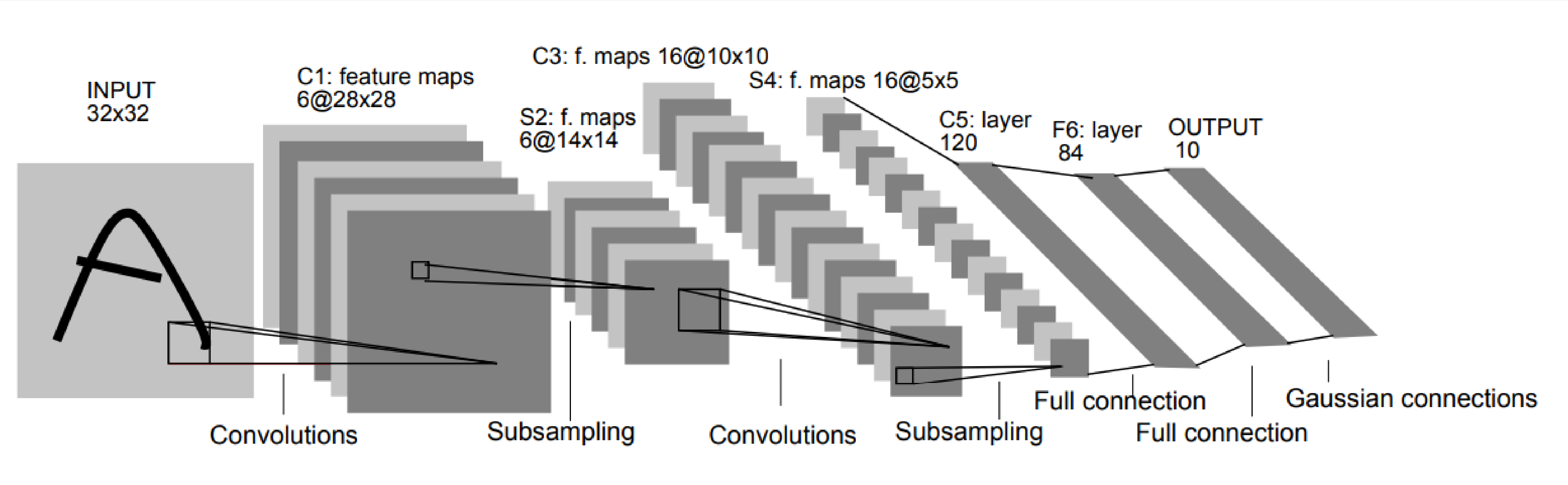
---By having less spatial information you gain computation performance

---Less spatial information also means less parameters, so less chance to over-fit

---You get some translation invariance

3 . Full connection layer

The CNNs help extract certain features from the image , then fully connected layer is able to generalize from these features into the output-space.



[LeCun et al.,1998.Gradient-based learning applied to document recognition.]

4 . Classic networks---AlexNet

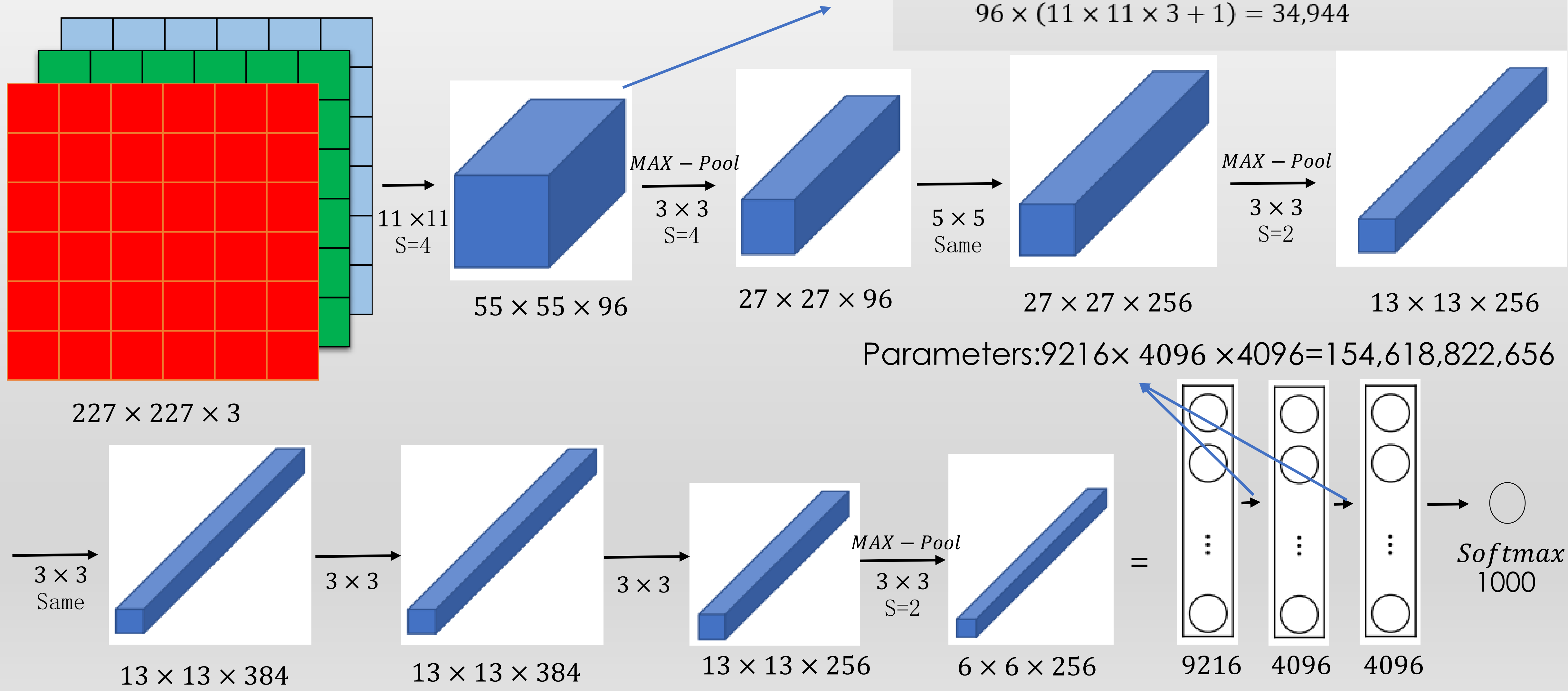
Parameters:

If Using Full connection layer:

$$55 \times 55 \times 96 \times (11 \times 11 \times 3 + 1) = 105{,}705{,}600$$

If using convolution:

$$96 \times (11 \times 11 \times 3 + 1) = 34{,}944$$



Thank you