



UNIVERSITAS ISLAM INDONESIA

The Fundamentals in Deep Learning

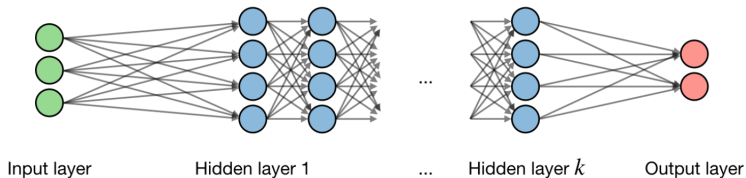
Ridho Rahmadi

Universitas Mataram

September 15, 2019

Deep Learning

Neural network



Let i be the i^{th} layer of the network and j the j^{th} hidden units (neurons) of the layer, we have

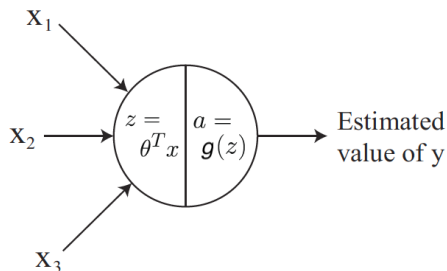
$$z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]},$$

where w, b, z are the weight, bias, output, respectively. The hidden neuron is called “hidden” because we don’t have the ground truth/training values.

(To conform with the standard neural network notation, we use w instead of θ)

Neural network

What is calculated in a single neuron? That is,



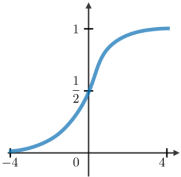
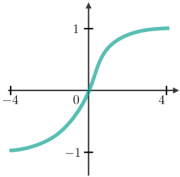
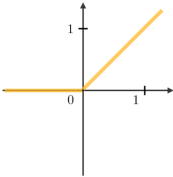
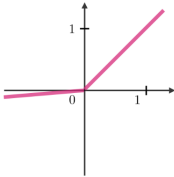
or equivalently,

$$z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]} \text{ and } a_j = g(z_j^{[i]}).$$

where g is a non-linear function (often called activation).

Activation function

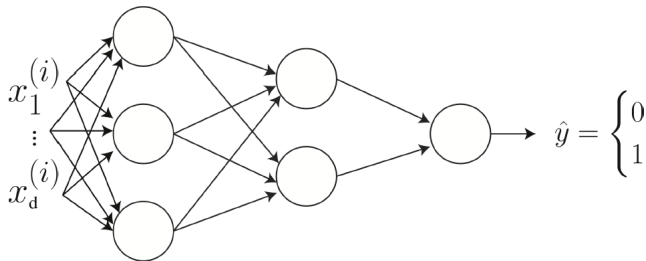
The activation functions simulate the neuron activation in our brain: the larger (and positive) value the more activated the neuron, and the less activated otherwise.

Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$
			

Technically, the activation function introduces non-linear complexities to the model.

Learning in a neural network

Consider a two layer neural network. On the left, the input is a flattened image vector. $x_1^{(1)}, \dots, x_d^{(i)}$.



Notice that all inputs are connected to all neurons in the next layer, which is called a *fully connected* layer.

Learning in neural network

Or equivalently below, the forward propagation,

$$z^{[1]} = W^{[1]}x^{(i)} + b^{[1]}$$

$$a^{[1]} = g(z^{[1]})$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

$$a^{[2]} = g(z^{[2]})$$

$$z^{[3]} = W^{[3]}a^{[2]} + b^{[3]}$$

$$\hat{y}^{(i)} = a^{[3]} = g(z^{[3]})$$

and now recall the learning procedure which is described earlier.

Learning in neural network

1. Pick initial values for parameters W and b , e.g., randomly draw from $N(0, 0.1)$.
2. After a single forward pass, e.g., $z^{[1]}$ until the $\hat{y}^{(i)}$ in the previous example, the cost function using e.g., mini-batch gradient descent

$$J_{mb} = \frac{1}{B} \sum_{i=1}^B L^{(i)}$$

where B is the number of examples in the mini-batch and $L^{(i)}$ is the loss for a single example computed via the cross-entropy function

$$L(y, \hat{y}) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})].$$

3. Backpropagate to update the model parameters
4. Repeat Steps 2 and 3 until converge

Backpropagation

A method to update the weights in the neural network by taking into account the actual output and the desired output. The derivative with respect to weight w is computed using chain rule

$$\frac{\partial L(z, y)}{\partial w} = \frac{\partial L(z, y)}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w}.$$

Thus the weight is updated via

$$w \leftarrow w - \alpha \frac{\partial L(z, y)}{\partial w},$$

where α is the learning rate.

Deep learning

Deep learning is based on neural network architecture, and that the learning procedure is basically the same as described above.

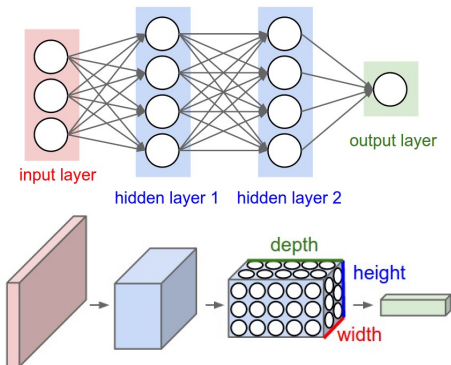
Supervised learning:

- ▶ CNN
- ▶ RNN
- ▶ LSTM

Unsupervised learning:

- ▶ Autoencoder
- ▶ Generative learning
 - ▶ Variational autoencoder
 - ▶ Generative Adversarial Network

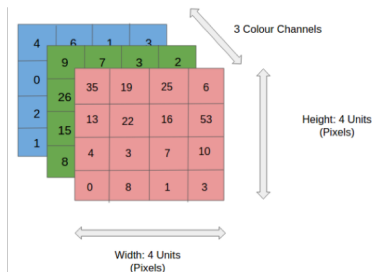
Convolutional Neural Network (CNN)



CNN is similar to the ordinary neural network.

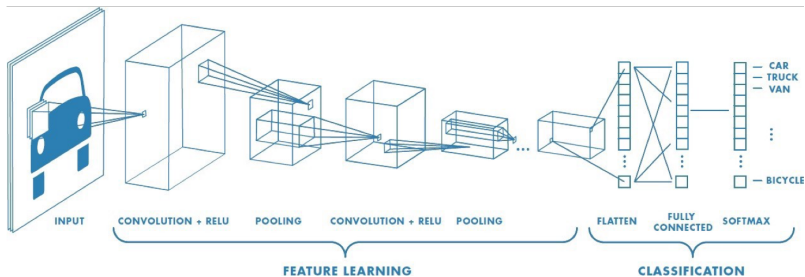
CNN

The difference is that CNN makes the explicit assumption that the inputs are images, allowing us to encode certain properties into the model.



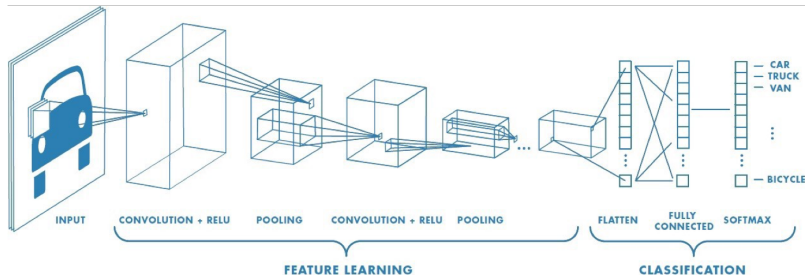
- ▶ A colored image consists of Red, Green, Blue (RGB) channels.
- ▶ In terms of matrix, a colored image is a matrix with 3 layers.
- ▶ See this example, an image of size $4 \times 4 \times 3$.

CNN



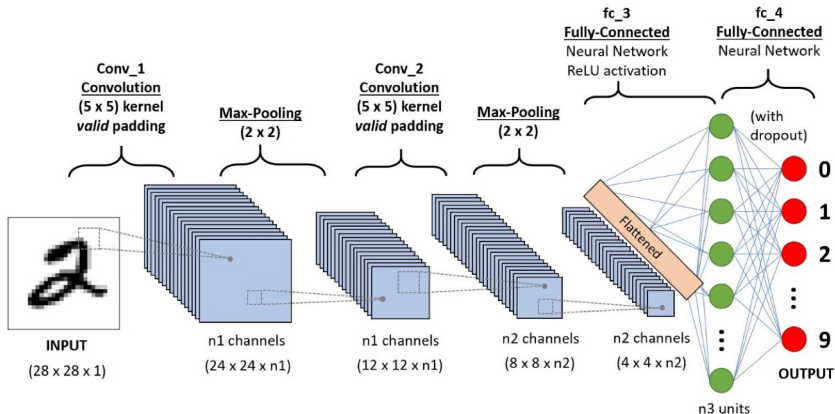
The idea of CNN is to reduce an image into a form that is easy to analyze, while preserving the most important features from the image.

CNN



- ▶ CNN is good at finding high-level features (e.g., nose, eyes patterns) from an image
- ▶ CNN develops the features along the layers
- ▶ At the final layer, the obtained features are used for further analysis, e.g., classification.

CNN example



Kernel

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

- ▶ A kernel or filter is used for detecting important features; orientation, nose, eyes, etc.
- ▶ A kernel is matrix and is moved by sliding (convolve)
- ▶ In every position, an element-wise multiplication is computed and then summed, resulting in convolved features

See this [link](#) for the animation.

Kernel

0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...

Input Channel #1 (Red)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1



308

+

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...

Input Channel #2 (Green)

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2



-498

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...

Input Channel #3 (Blue)

0	1	1
0	1	0
1	-1	1

Kernel Channel #3



164

+ 1 = -25

Bias = 1

Output

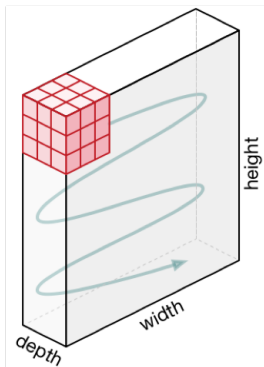
-25			...
			...
			...
			...
...

See this [link](#) for the animation.

Kernel

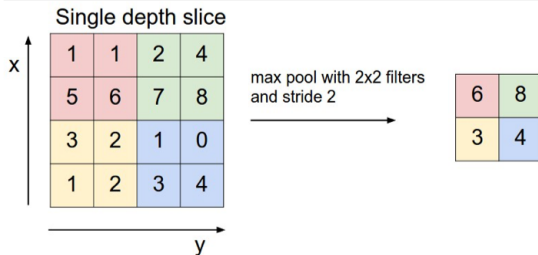
See <http://cs231n.github.io/convolutional-networks/>

Stride



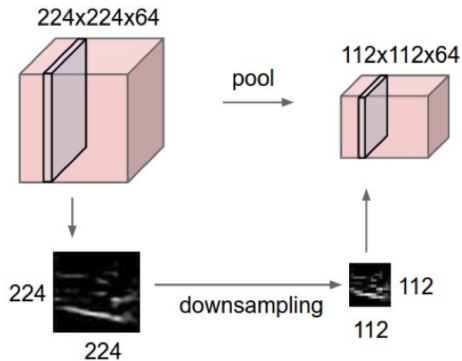
- ▶ The kernel size is typically $3 \times 3 \times n$ or $5 \times 5 \times 5$
- ▶ The parameter **stride** controls the slide of the kernel, e.g., $\text{stride} = 1$ meaning that we move the kernels one pixel at a time

Pooling

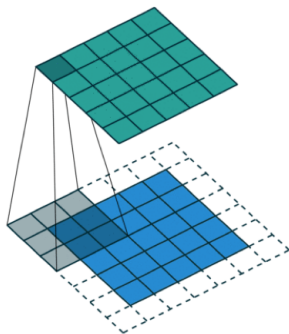


- ▶ Pooling is used to reduce complexity from the convolved layer.
- ▶ This means, to reduce the parameter number is a CNN; to avoid overfitting.
- ▶ Max pooling is often used, that is, it selects the maximum value, given a pooling filter.

Max Pooling



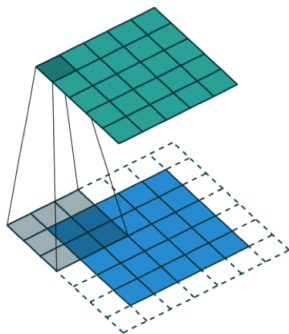
Padding



- ▶ (Zero) Padding, is *framing* the image with , typically zero's
- ▶ Padding is to help recognizing pattern from the image

See this [link](#) for the animation.

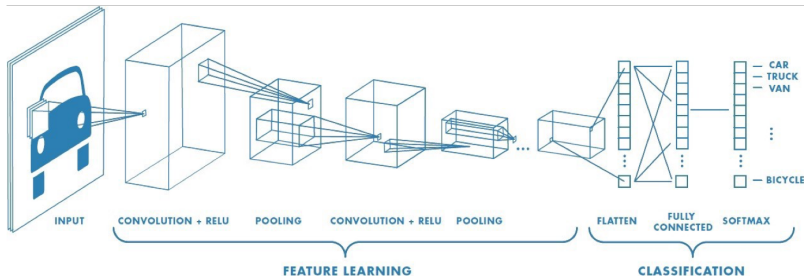
Padding



See this [link](#) for the animation.

- ▶ Same Padding (see the example); resulting convolved matrix has the same size of that the input.
 - ▶ See the example: input size is 5×5 , with padding becomes $7 \times 7 \times 1$. Using a kernel of $3 \times 3 \times 1$ results in a convolved matrix of $5 \times 5 \times 1$.
- ▶ Valid Padding or without padding will result in a convolved matrix of size the same with that of the kernel.

CNN



After CNN learns the features, the next phase is to classify the features.

Sources

- ▶ Andrew Ng's machine learning materials
- ▶ An Introduction to statistical learning, James et al.
- ▶ <https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-deep-learning>
- ▶ <http://cs231n.github.io/convolutional-networks/>
- ▶ <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- ▶ <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>