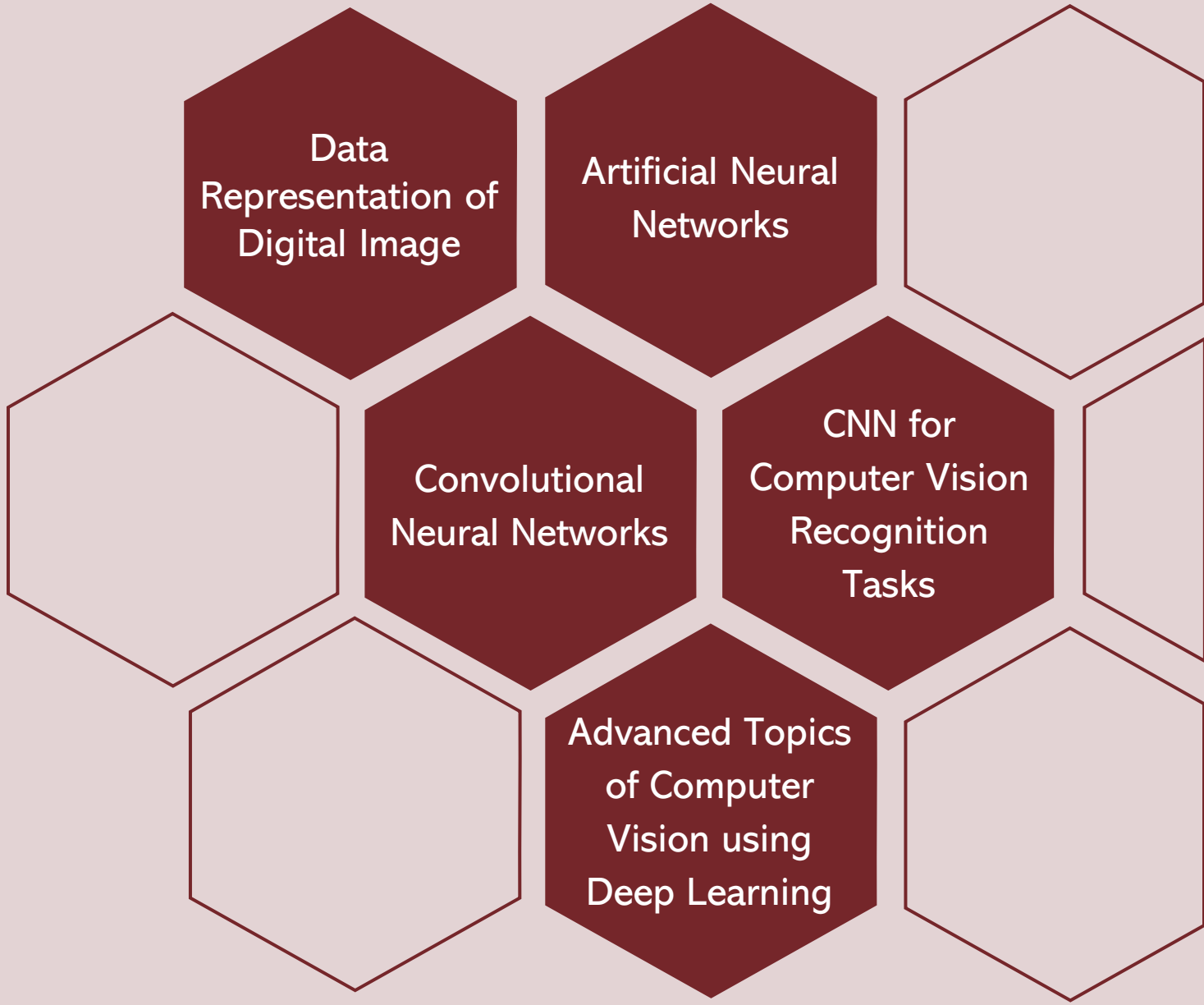
Semester Gasal 2024/2025
Topic #11

Neural Networks for Computer Vision

CSCE604133 Computer Vision
Faculty of Computer Science
Universitas Indonesia

Dr. Eng. Laksmita Rahadianti
Muhammad Febrian Rachmadi, Ph.D.
Dr. Dina Chahyati, Prof. Dr. Aniati M. Arymurthy

Agenda

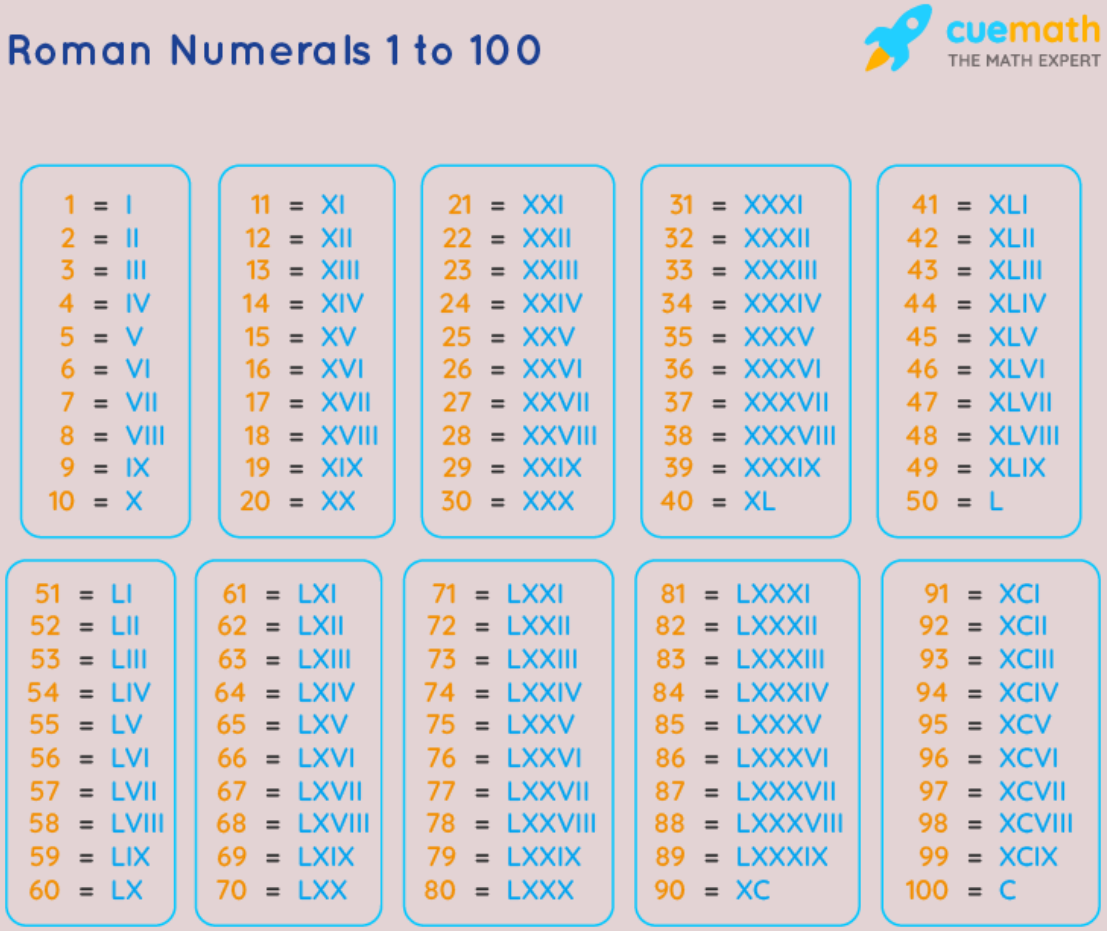

Data Representation of Digital Image
Artificial Neural Networks
Convolutional Neural Networks
CNN for Computer Vision Recognition Tasks
Advanced Topics of Computer Vision using Deep Learning

Data Representation of Digital Image

Section 1

Data Representations in Our World

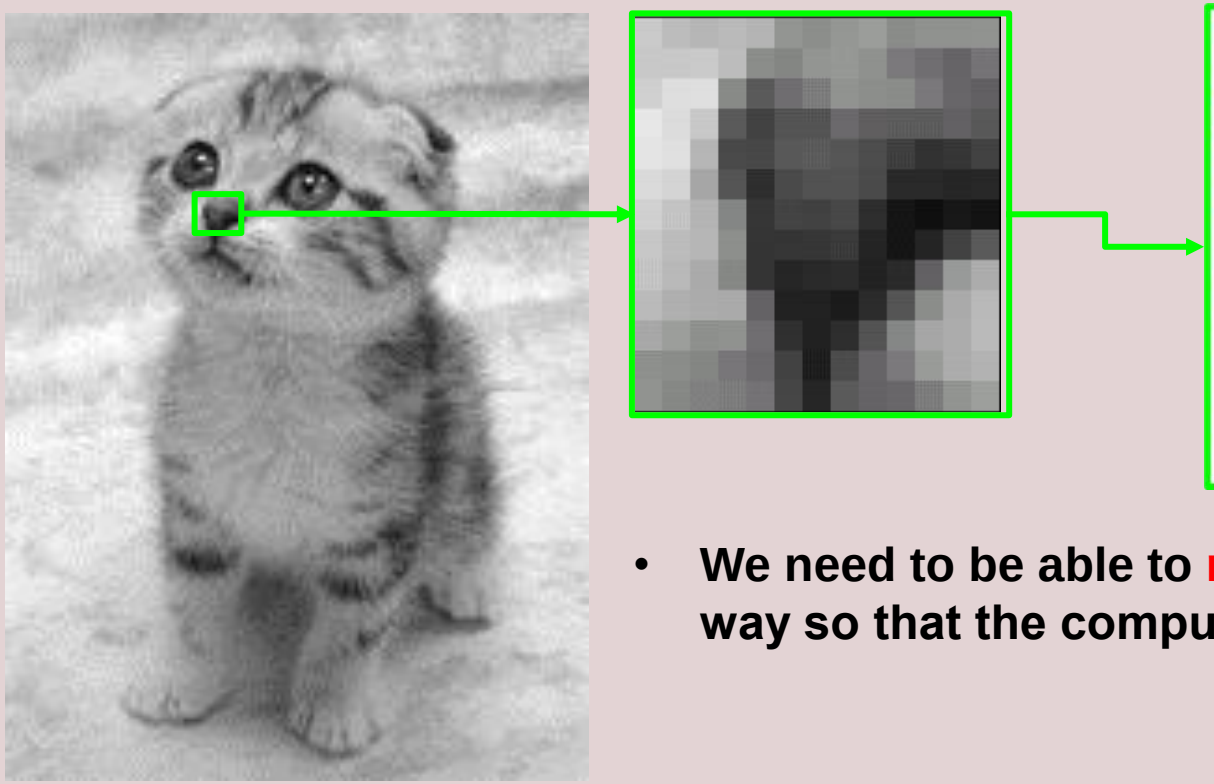
- Data can be represented differently but still have the same meaning.
- An easy example is the representation of numbers, which have various representations such as Roman, Arabic (decimal), hexa-decimal, binary, etc.
- For modern humans, Arabic (decimal) numbers are much easier to calculate than other number representations such as Roman and binary.
- However, for computers, binary is much easier to calculate and use than other representations, especially in mathematical calculations.



https://www.cuemath.com/numbers/roman-numerals-1-to-100/

The Difficulty of Learning from Digital Data for Human (e.g., in Computer Vision)

- What humans see



- What computers see

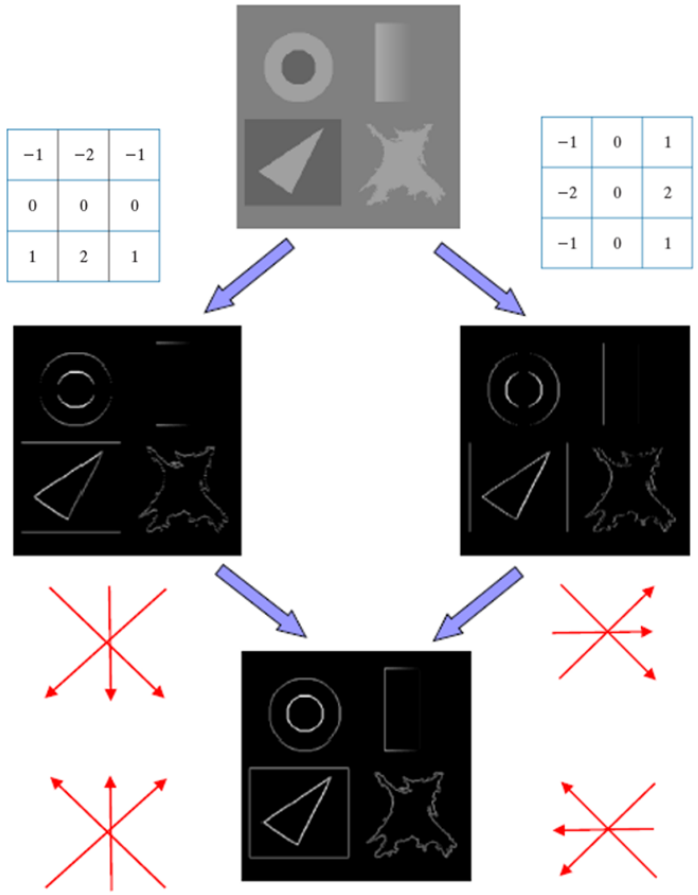
201	188	181	185	180	147	140	149	155	138	144	144	145
199	200	201	188	139	132	147	150	143	123	112	102	117
207	221	222	136	90	111	125	145	140	138	122	104	97
231	219	200	90	65	84	84	107	95	92	92	99	89
227	223	181	74	72	89	92	86	77	63	50	55	65
217	211	166	85	47	75	82	83	75	42	42	39	40
208	195	179	131	54	68	66	72	46	21	15	24	19
198	187	181	141	53	54	55	59	37	21	37	66	90
195	184	170	134	52	38	42	45	35	43	98	152	172
186	175	171	169	100	34	34	27	44	85	139	170	184
167	156	142	144	112	48	32	46	84	133	166	172	186
142	139	131	120	108	67	30	76	102	123	153	171	178
145	134	128	125	117	70	38	91	101	105	125	146	157

- We need to be able to **represent** the image in a certain way so that the computer knows what it's looking at.

Gonzalez, R. C. (2009). *Digital image processing*. Pearson education india.
Pengolahan citra course, Fasilkom UI by A. Murti, M.F. Rachmadi, and L. Rahadianti

Feature Extraction for Better Representation

- Feature extraction is one way for us to create a better data representation of an image.
- Filters/kernels are the easiest and fastest way to extract features from an image.

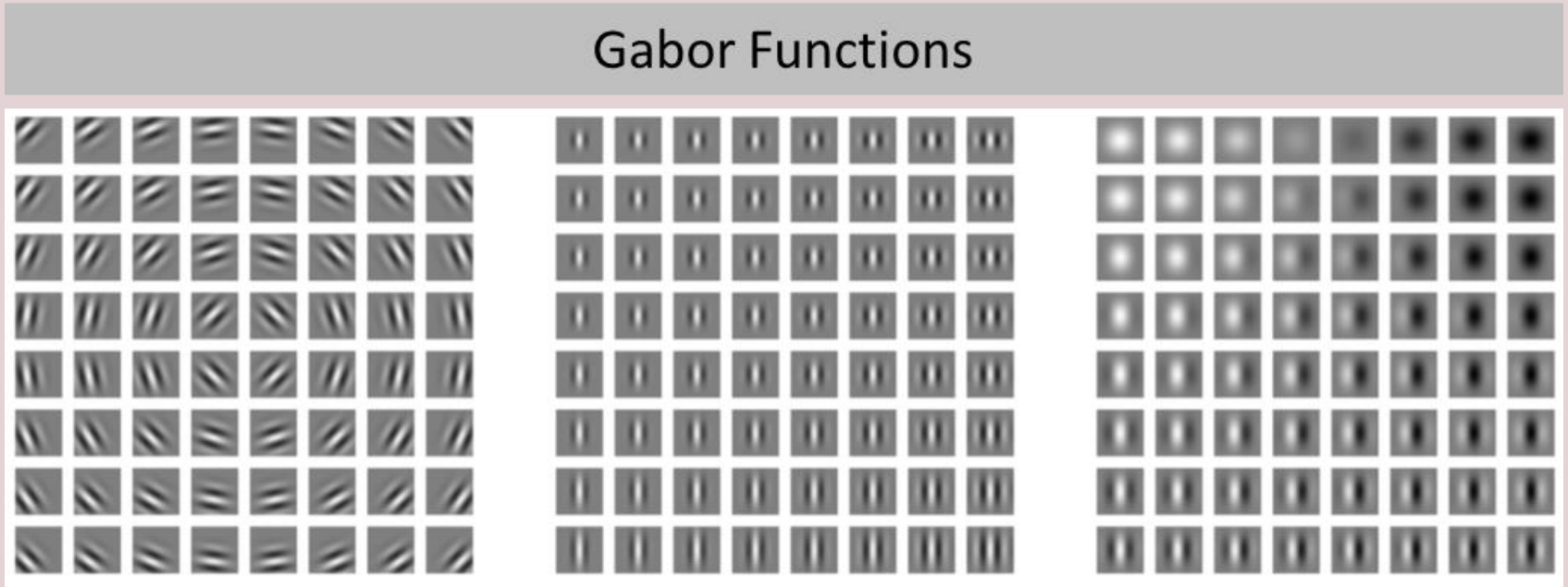


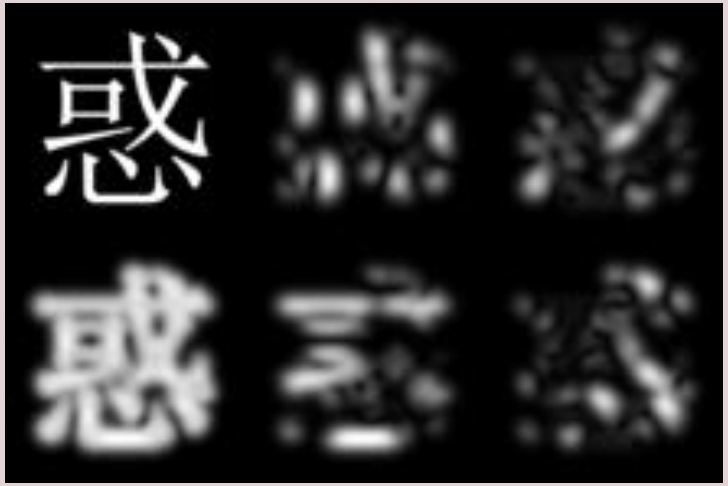
- **An example:** Sobel gradient operators for edge detection.
- **Problems:**
  - Edge is a low-level feature that is not meaningful to humans.
  - We can extract lines or circles from edges, but these are middle-level features that might be meaningful or not.
  - How about high-level features?

Gonzalez, R. C., and R. E. Woods. "Digital Image Processing, 4th Global Edition."

Gabor Filters for Feature Extraction

- Each feature extraction has a set of specific filters/kernels for extracting features (edges) from an image.
- We can create even more complex features from a set of complex filters/kernels (e.g., **Gabor filters**).
- Gabor filters are defined by a sinusoidal wave (a plane for 2D filters) multiplied by the Gaussian function.





Demonstration of a Gabor filter applied to Chinese OCR.

- However, even the more complex Gabor filters cannot extract high-level features of an image.

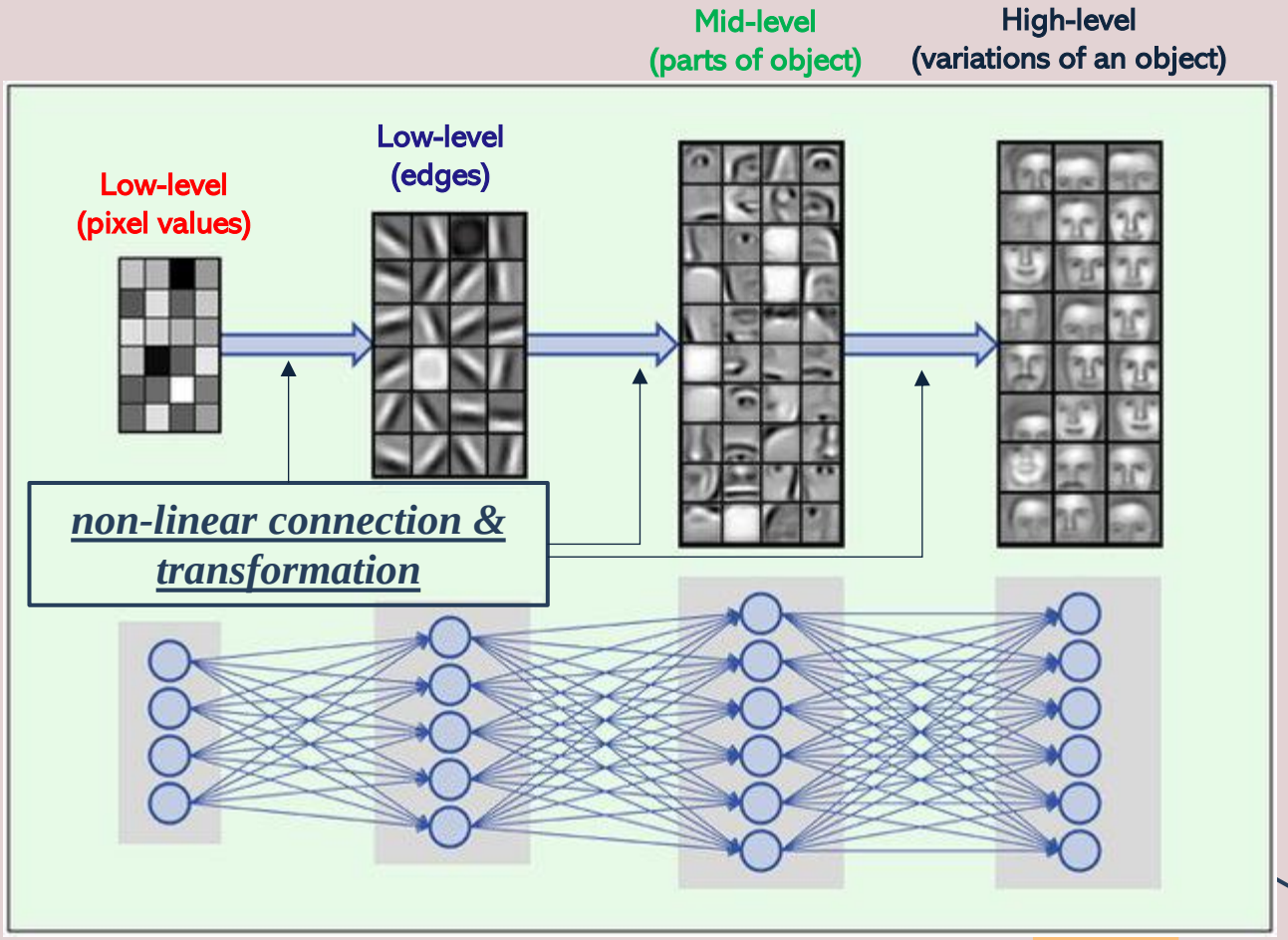
https://en.wikipedia.org/wiki/Gabor_filter

Representation Learning in Computer Vision using Convolutional Neural Networks (CNN)

=

- **The problem:** Except for the low-level features (e.g., edges), it is very difficult to extract **high-level**, **abstract features**, and **factors of variations** from raw data.

- **CNN and deep learning** solve this central problem by introducing representations that are expressed hierarchically.



Mayr, Andreas, et al. "DeepTox: toxicity prediction using deep learning." *Frontiers in Environmental Science* 3 (2016): 80.

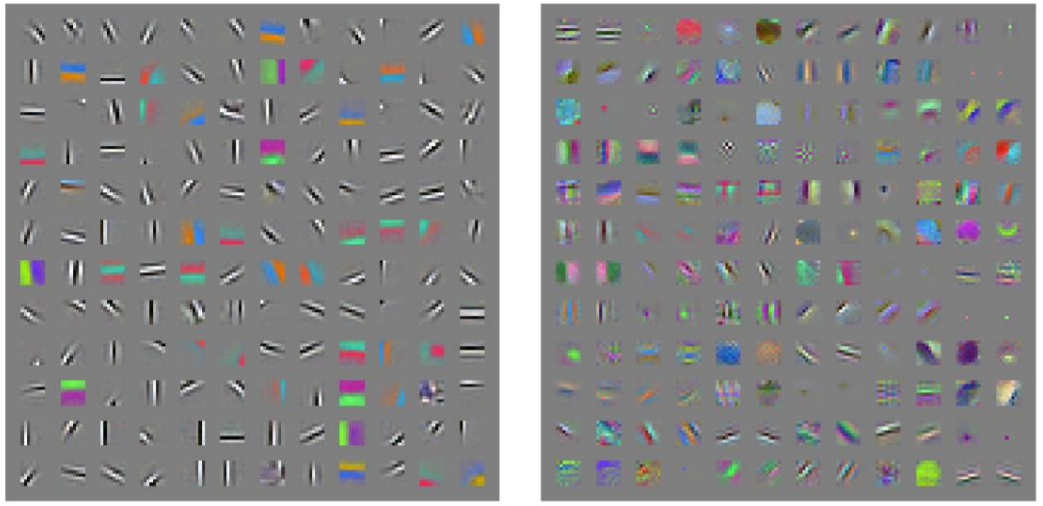
Why is the CNN Powerful?

CNN Powerful karena kalau kita punya rule based system,

mereka terlibat dari awal untuk mengklasifikasikan output

- **Shaded boxes** indicate components that are automatically learned from data to automatically solve the non-linearity connections of our problem.
- After the training process, CNN usually produces **Gabor-like filters**. The trained filters give us the best insight into what features were taken/learned directly from the training images.

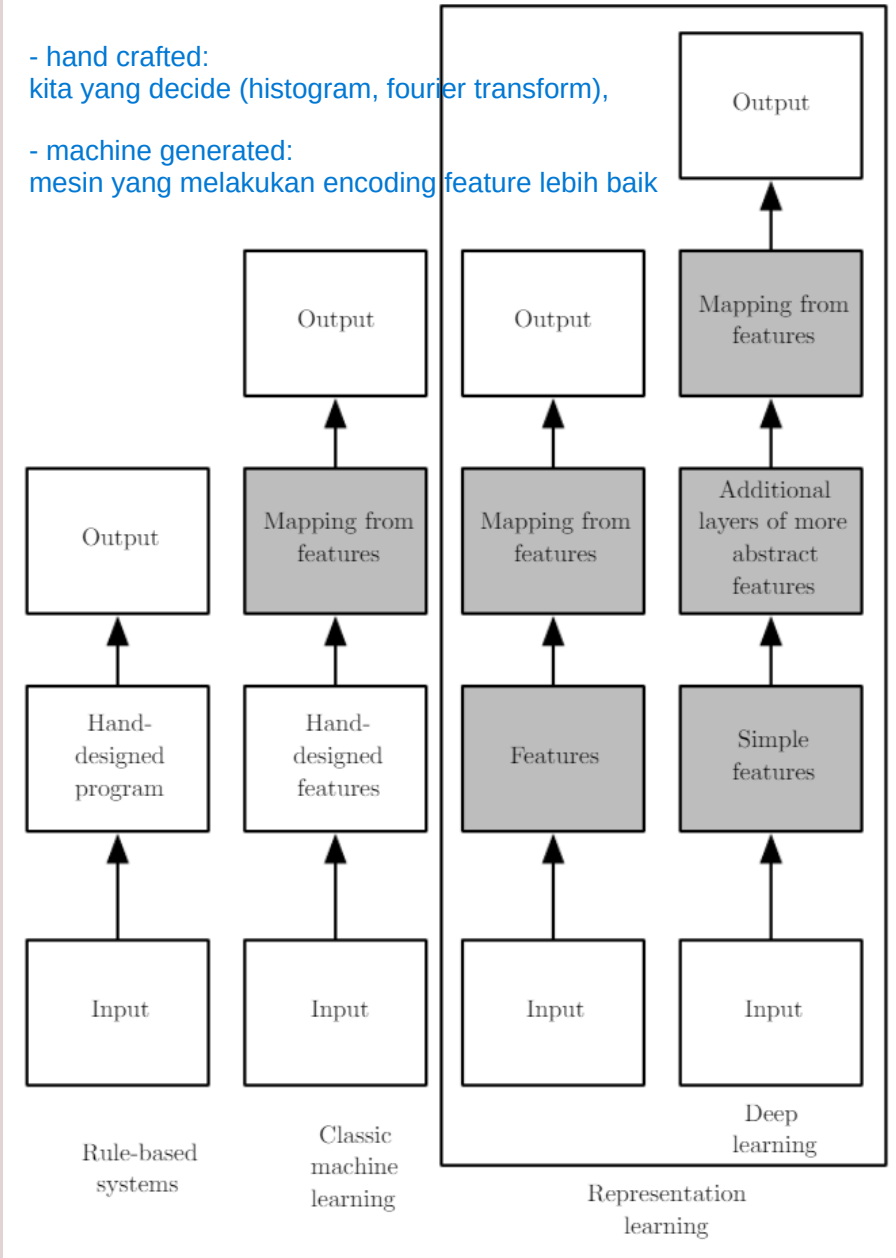
Gabor-like Learned Kernels from CNN



Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

- hand crafted:
kita yang decide (histogram, fourier transform),

- machine generated:
mesin yang melakukan encoding feature lebih baik



Neuroscience behind the CNN

- David H. Hubel and Torsten Wiesel performed a series of experiments on cats in 1958 and 1959 and on monkeys several years later, giving crucial insights into the structure of the visual cortex. The authors received the Nobel Prize in Physiology or Medicine in 1981 for their work.
- Some important outcomes from these experiments are:
  1. They showed that some neurons react only to images of horizontal lines, while others react only to lines with different orientations.
  2. They showed that many neurons in the visual cortex have a small local receptive field, meaning they react only to visual stimuli located in a limited region of the visual field (i.e., low-level features).
  3. They also noticed that some neurons have larger receptive fields, and they react to more complex patterns that are combinations of the lower-level patterns (i.e., high-level features).
  4. These observations led to the idea that the higher-level neurons are based on the outputs of neighboring lower-level neurons (i.e., low-level and high-level processing are connected to each other from low-level to high-level features).
- This powerful architecture is able to detect all sorts of complex patterns in any area of the visual field.

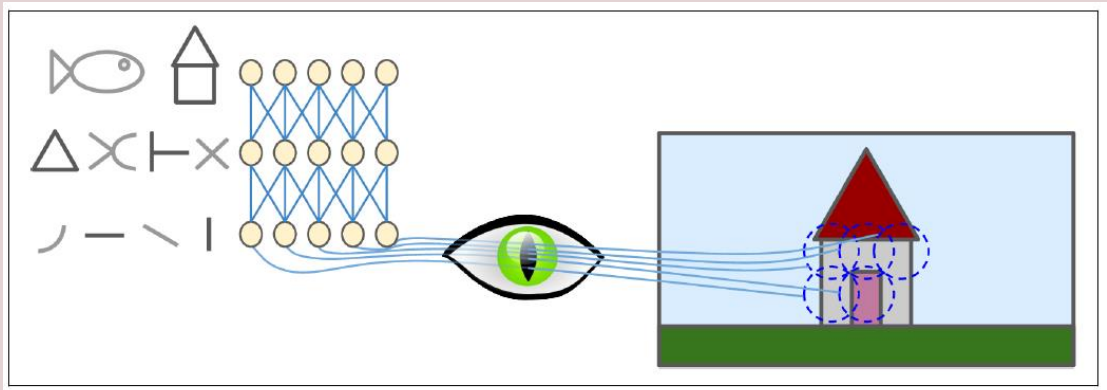
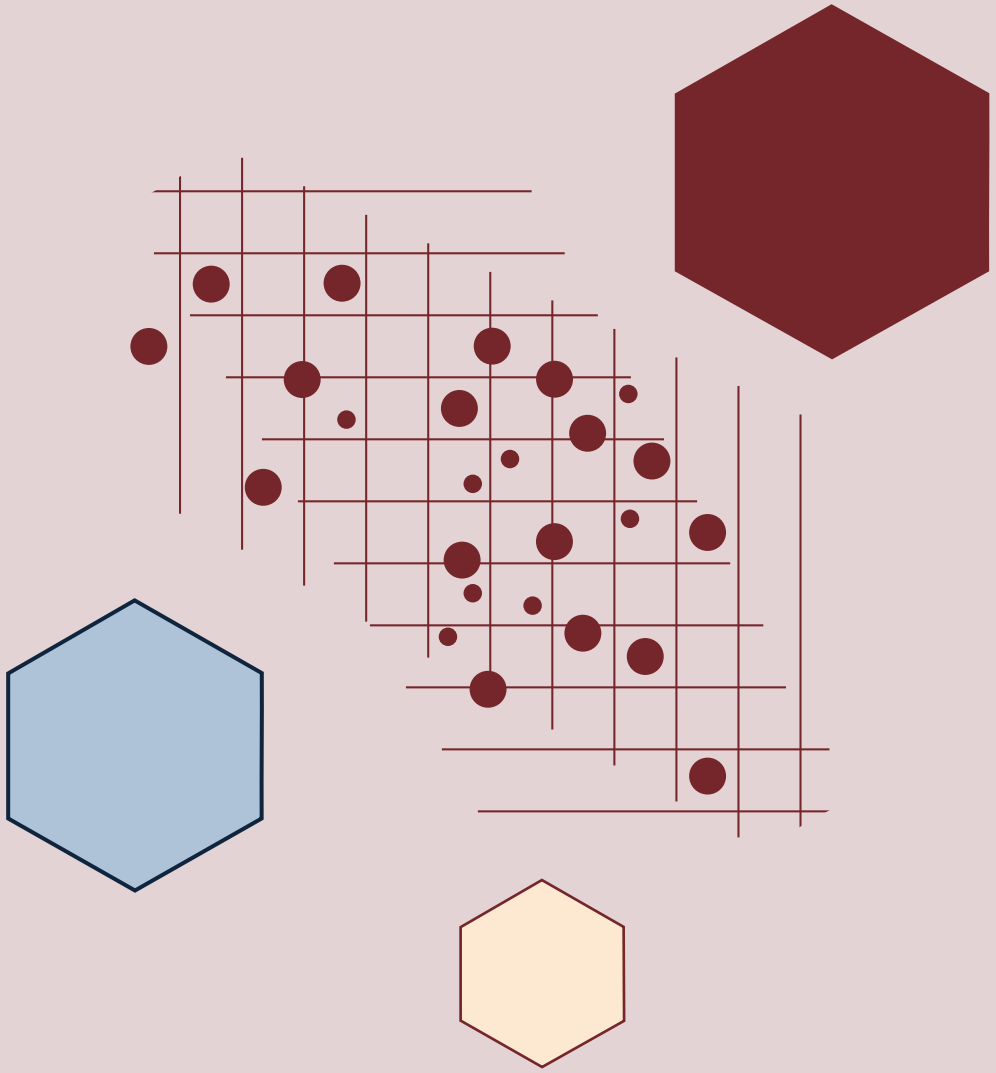


Figure 14-1. Local receptive fields in the visual cortex

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.



Artificial Neural Networks

Section 2

Biological Neural Networks

A Single Neuron

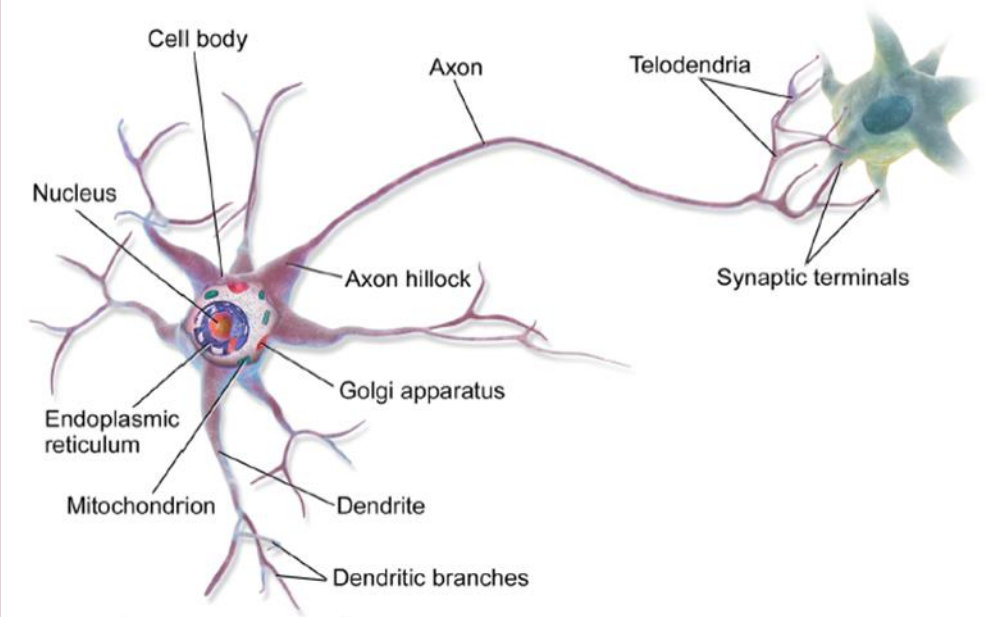


Figure 10-1. Biological neuron[3]

3 Image by Bruce Blaus (Creative Commons 3.0). Reproduced from *https://en.wikipedia.org/wiki/Neuron*.

Connections of Neurons

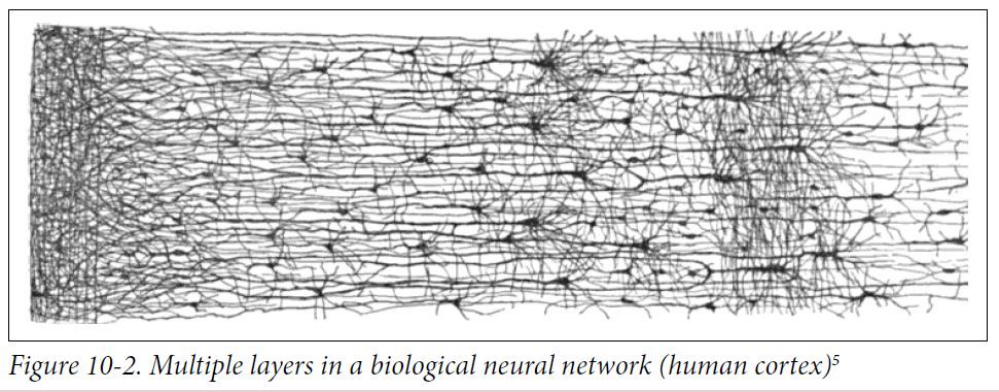
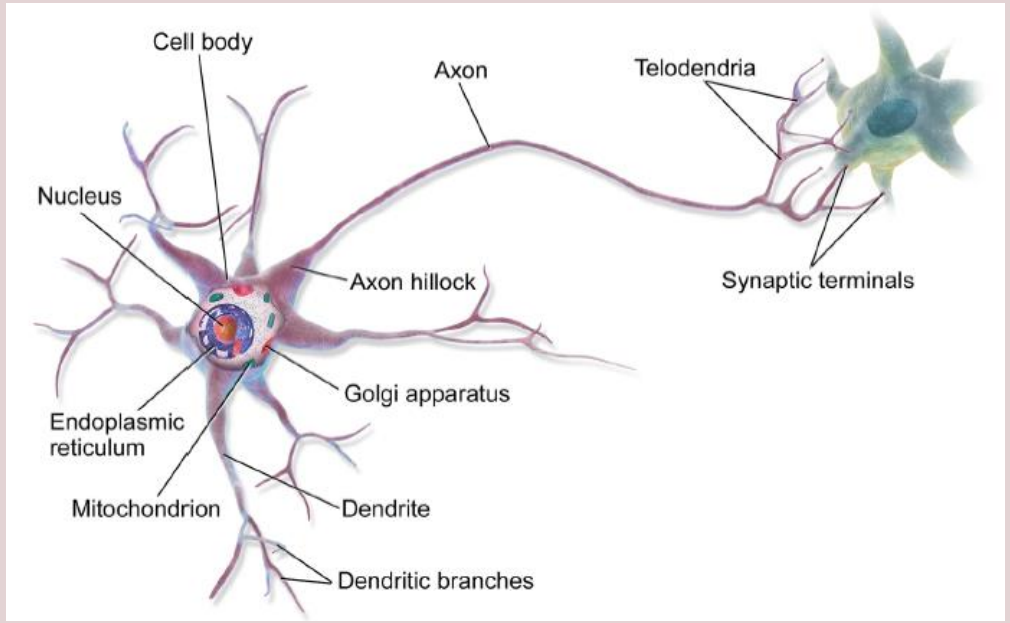
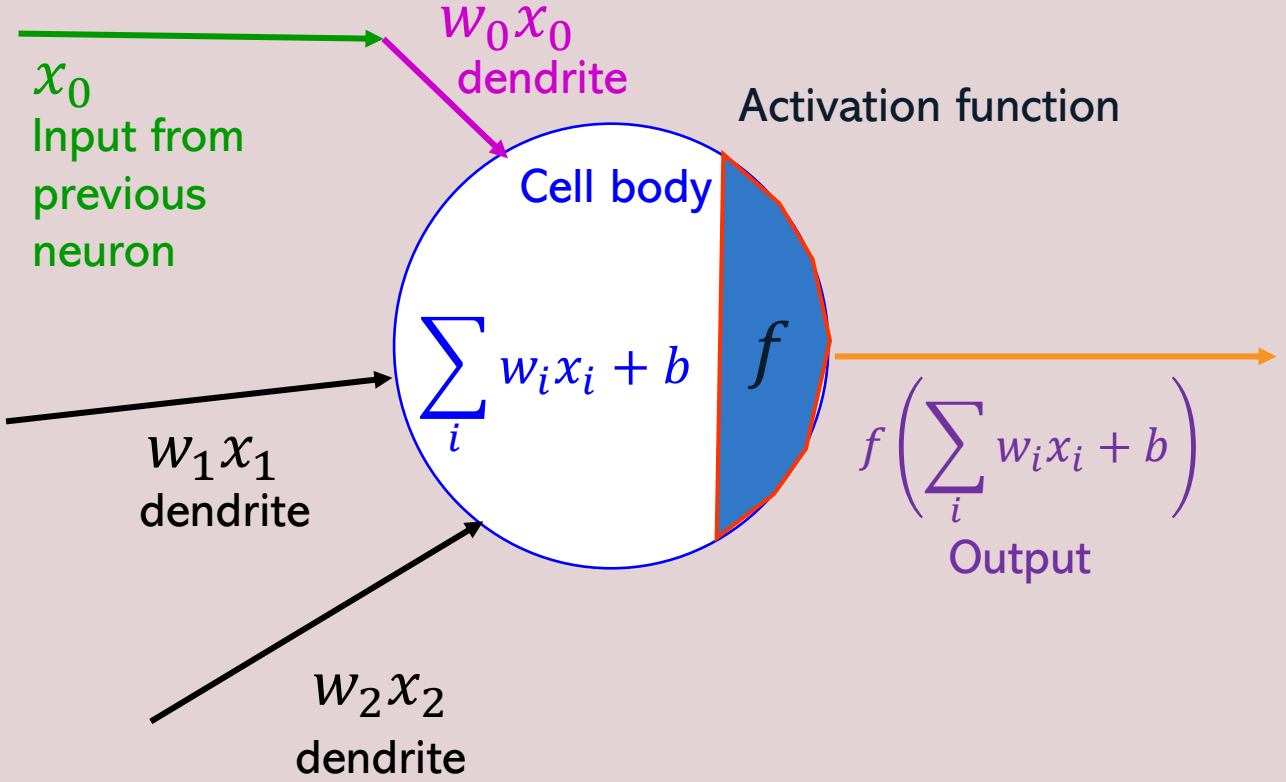


Figure 10-2. Multiple layers in a biological neural network (human cortex)[5]

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Biological vs Artificial Neuron

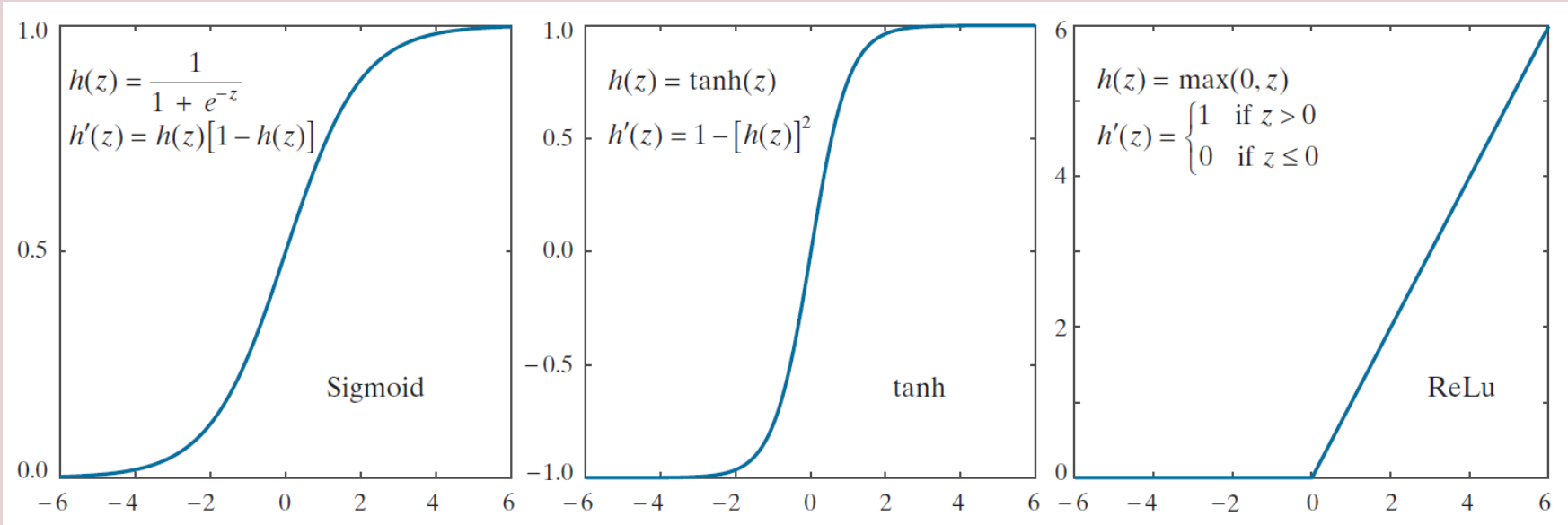




CS231n Convolutional Neural Networks for Visual Recognition: http://cs231n.github.io/

Non-Linear Activation Function

- Activation functions **ensures non-linearity**
- A series of linear transformations only result in another linear transformation
- Without non-linearity, the stack of layers would be equal to a single layer



Sigmoid

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

Tanh

$$tanh(x) = 2\sigma(2x) - 1$$

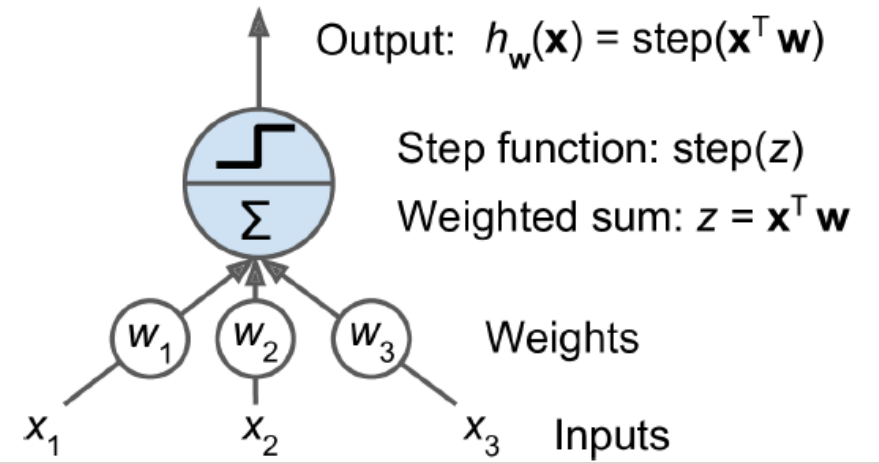
ReLU
(Rectified Linear Unit)

$$f(x) = \min(0, x)$$

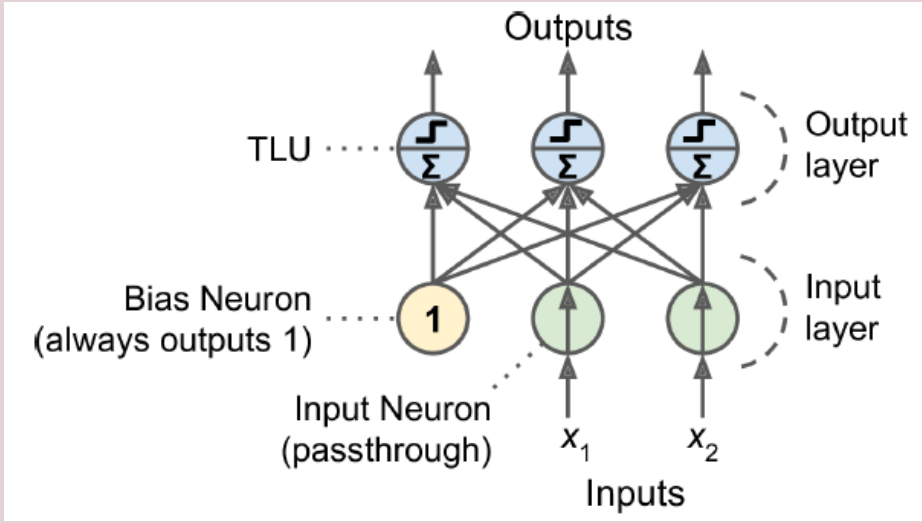
The Simplest Form of ANN

cuman 1 fitur aja (iterasi) termasuk ANN

- One of the simplest artificial neural networks was invented in 1957 by Frank Rosenblatt, which is based on threshold logic unit (TLU) proposed by McCulloch, W.S. and Pitts, W., 1943.



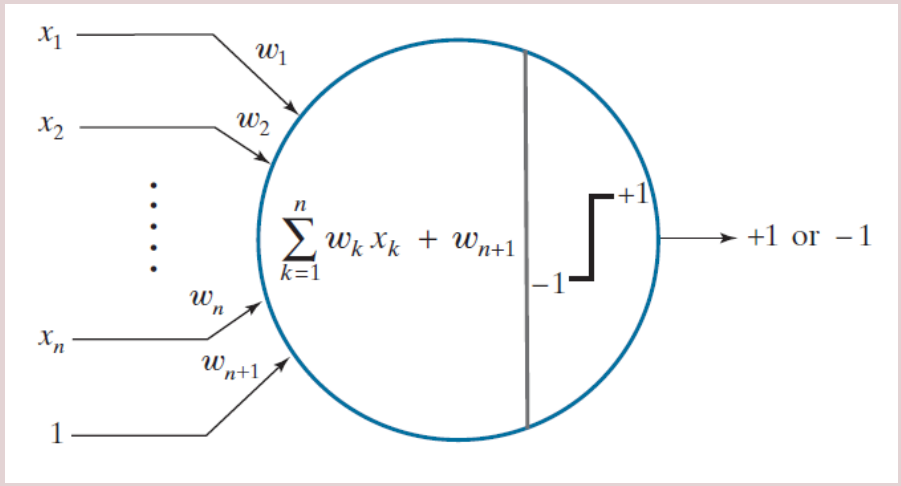
McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115-133.



Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

The Perceptron

- Rosenblatt showed mathematical proofs showing that the perceptron, when trained with linearly separable training sets (i.e., training sets separable by a hyperplane), would converge to a solution in a finite number of iterative steps.



Perceptron learning rule (weight update)

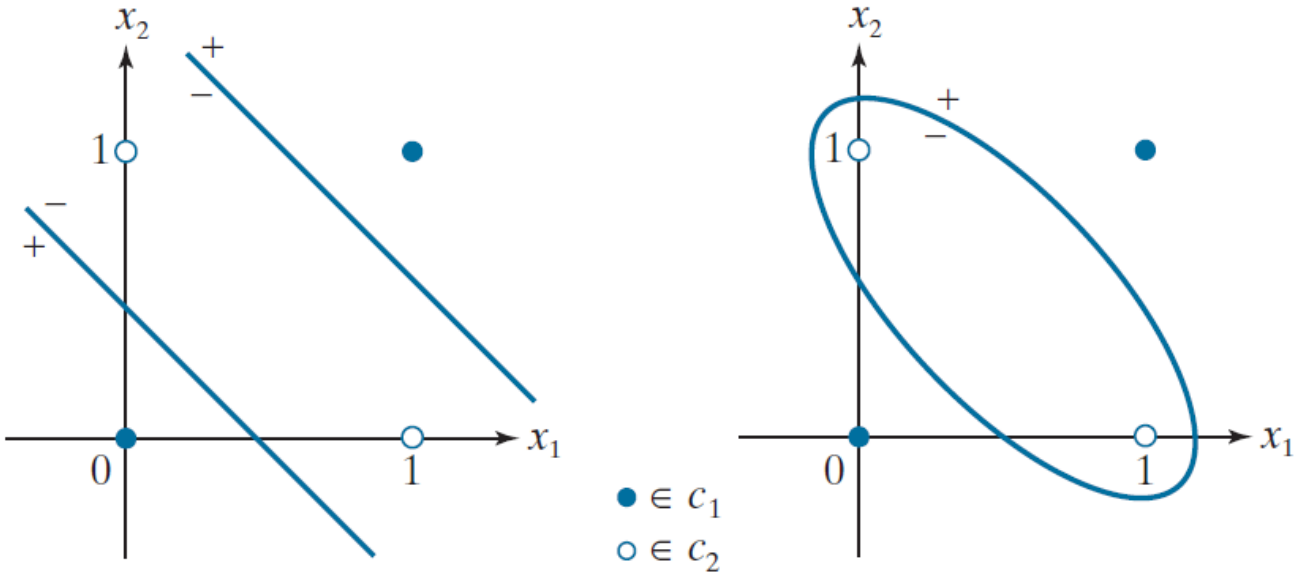
$$w_{i,j}^{(next\ step)} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

- $w_{i,j}$ is the connection weight between the i^{th} input neuron and the j^{th} output neuron.
- x_i is the i^{th} input value of the current training instance.
- $\hat{y}_j$ is the output of the j^{th} output neuron for the current training instance.
- y_j is the target output of the j^{th} output neuron for the current training instance.
- η is the learning rate.

Serious Weaknesses of the Perceptron

- Perceptron can not solve some trivial problems, e.g
  - Exclusive OR (XOR) classification problem
  - ** XOR is not linearly separable
- This can be solved by adding hidden layers, which results in a *multi-layer perceptron*.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

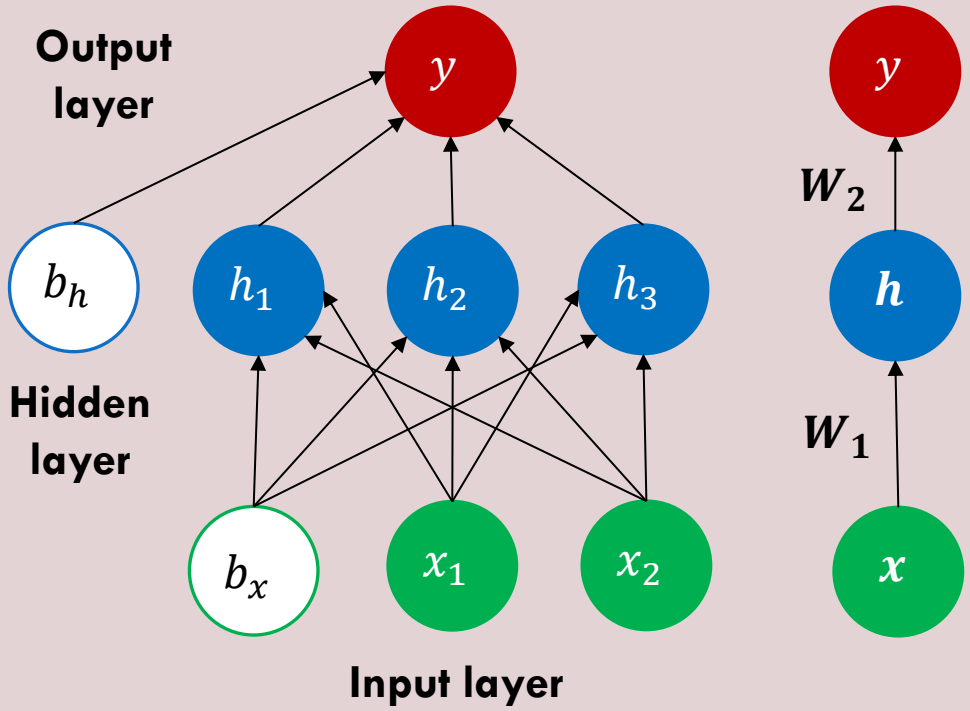


Gonzalez, R. C., and R. E. Woods. "Digital Image Processing, 4th Global Edition."

Multi-Layer Perceptron (MLP)

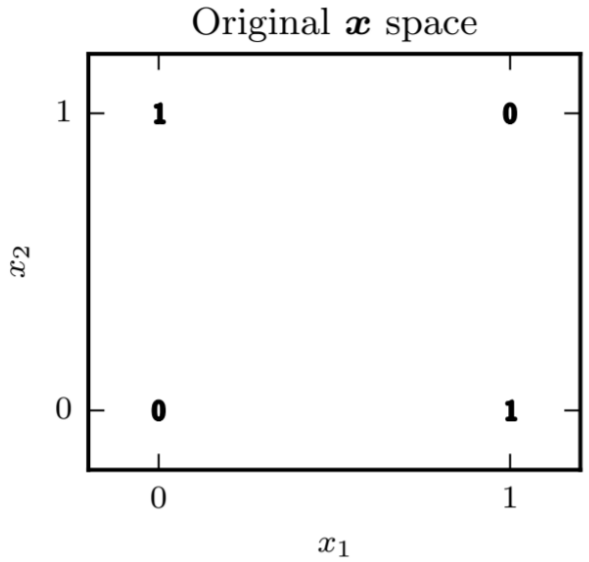
karena komputer kita bisa komputasi banyak layer, nah makanya layernya ditambahin aja.

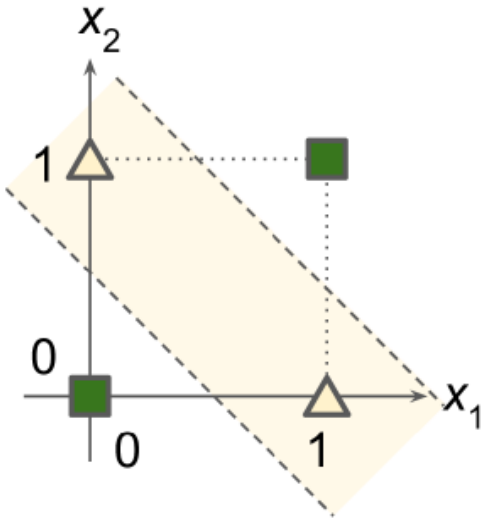
Tapi drawbacknya: makin dalam manusaia makin gak ngerti apa yang dilakukan di MLP ini.

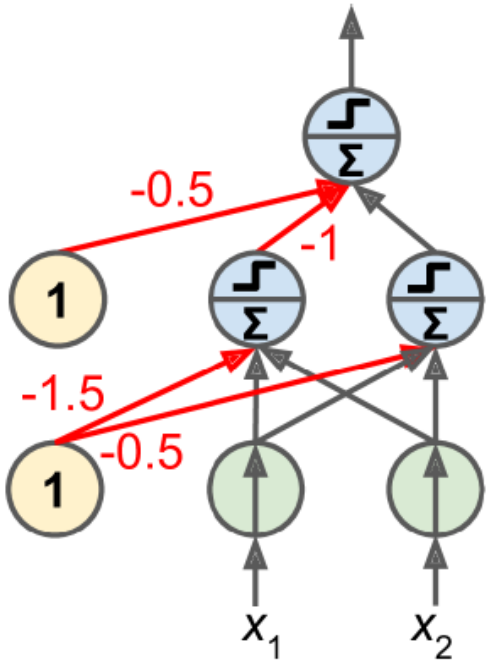


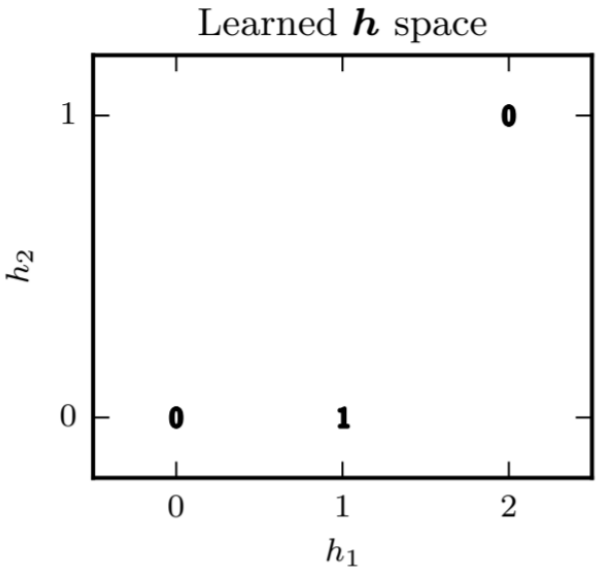
- Notation:
  - $\boldsymbol{x}$ represents the input features (as a vector).
  - The weight matrices $\boldsymbol{W_i}$ contains all connection weight, except from the bias neuron.
  - $\boldsymbol{h}$ represents the hidden layer's values.
  - ϕ is the activation (non-linear) function.
- Equation:
  - $\boldsymbol{h}_{\boldsymbol{W_1},b_x}(x) = \phi(\boldsymbol{x}\boldsymbol{W_1} + b_x)$
  - $y_{\boldsymbol{W_2},b_h}(h) = \phi(\boldsymbol{h}\boldsymbol{W_2} + b_h)$

MLP for Solving XOR





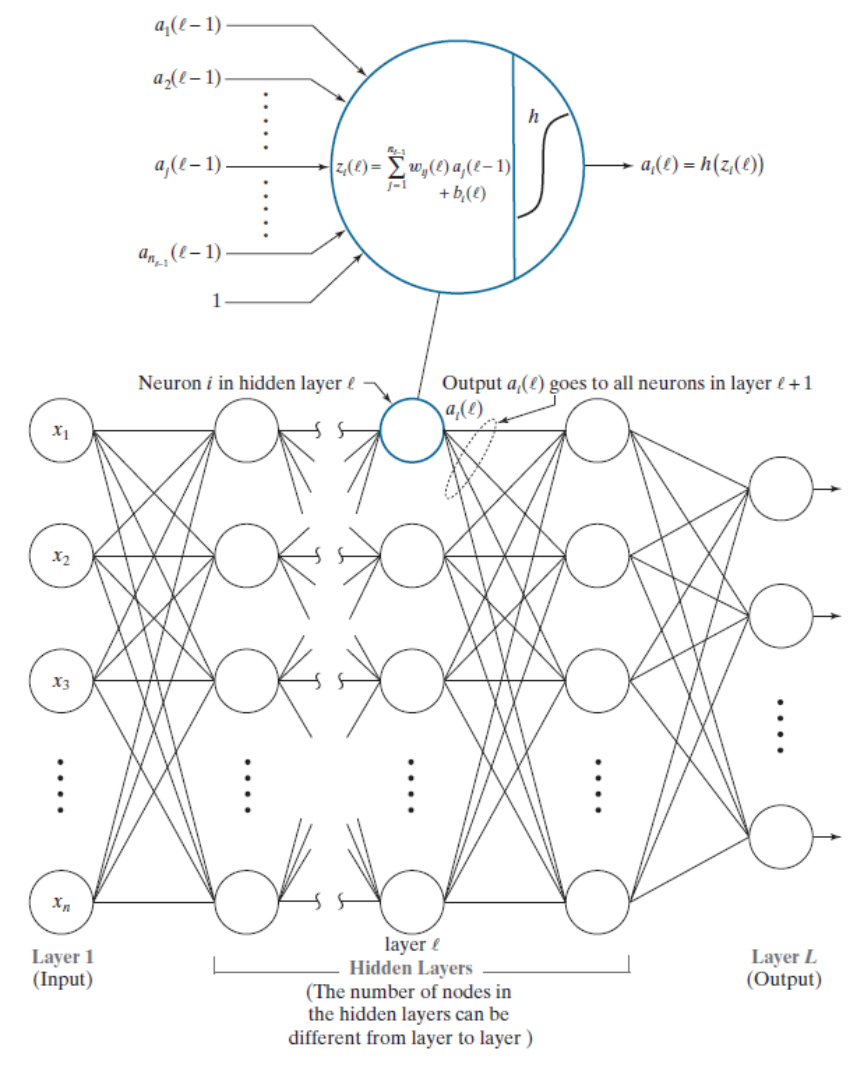




Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

MLP & Deep Neural Networks



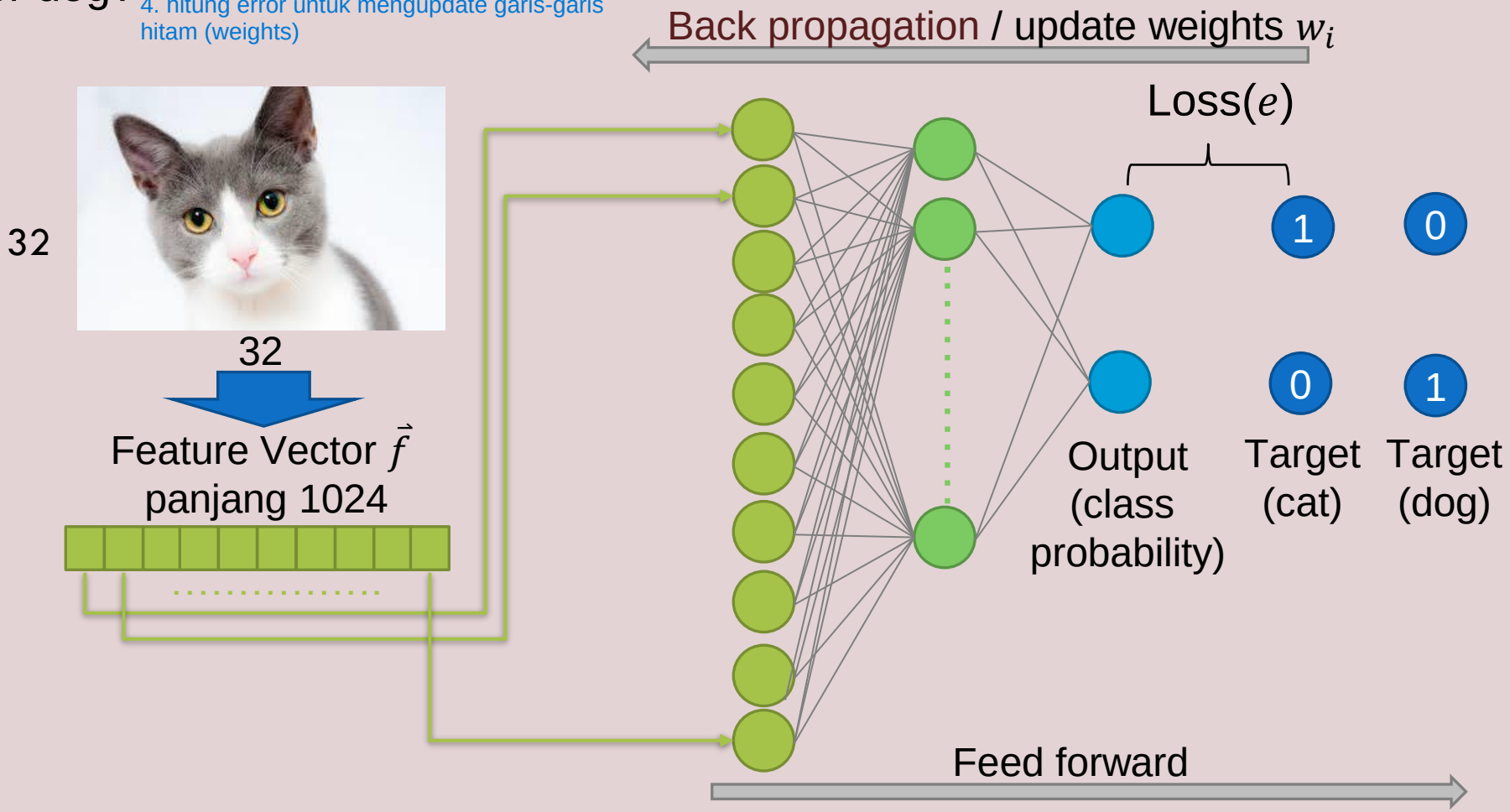
- MLPs can be trained with backpropagation and gradient descent
  - (+) We can determine the number of neurons and layers
  - (-) So many hyperparameters
- Deep neural networks are a deeper MLP (more later) with some additional techniques

Gonzalez, R. C., and R. E. Woods. "Digital Image Processing, 4th Global Edition."

Simple ANN for Image Classification

- Cat or dog?

1. mengubah gambar jadi feature vector
2.. memasukkan feature vector ke NN
3. Diulang lalu mendapatkan output.
4. hitung error untuk mengupdate garis-garis hitam (weights)



Backpropagation

- For each training instance (or mini-batch):

Forward pass: Each mini-batch is fed to the networks input layer, which then forwards it to the following layers until the last prediction in the output layer.

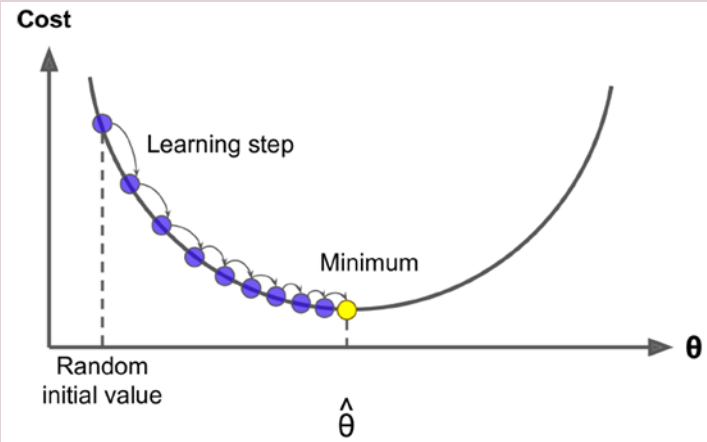
Error measurement: Measure the error between the output layer result with the ground truth.

Reverse pass: Go back through the layers to compute the error distribution of each connection

Gradient descent step: Adjust the connection weights to minimize the error.

Minimization as Optimization

- Minimizing the loss/cost/error function $f(x)$ using its derivative.
  - For (image) classification: (binary) cross entropy, etc.
  - For regression: mean square error (MSE), etc.
  - For (image) segmentation: cross entropy, Dice similarity coefficient (DSC), focal loss, etc.



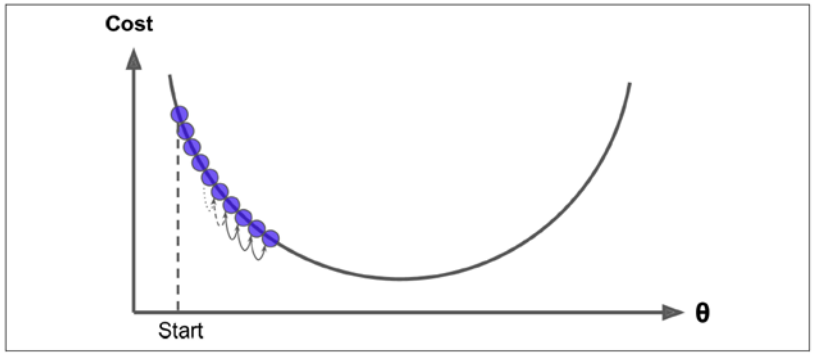


Figure 4-4. Learning rate too small

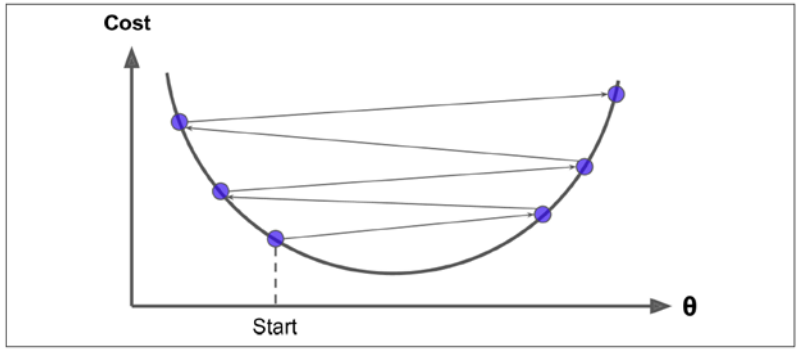


Figure 4-5. Learning rate too large

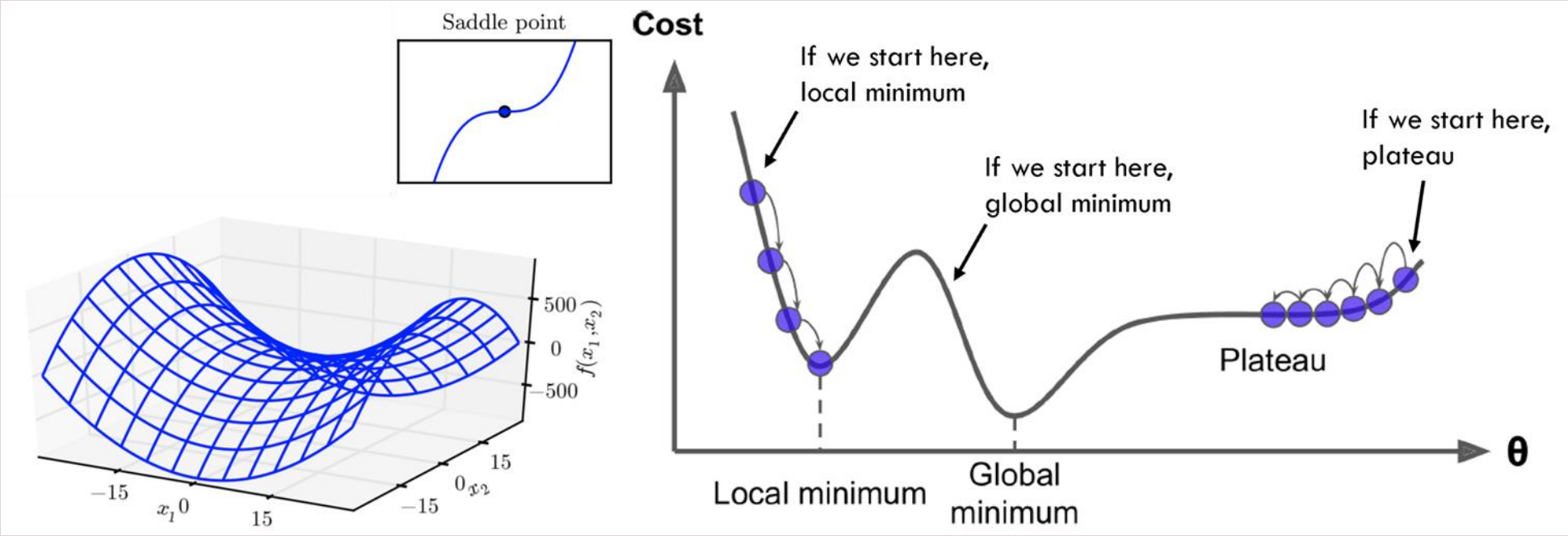
Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Weaknesses: Local Minima

kalau kita mulai dari titik random, bisa aja masuk ke local minima

meanwhile kita mau global minima

- In general, neural networks are hard to train/optimize.
  - Neural networks have a lot of ***saddle points***) and ***local minima***.
  - **Solution:** Better weights initialization (e.g., Glorot/He initialization).

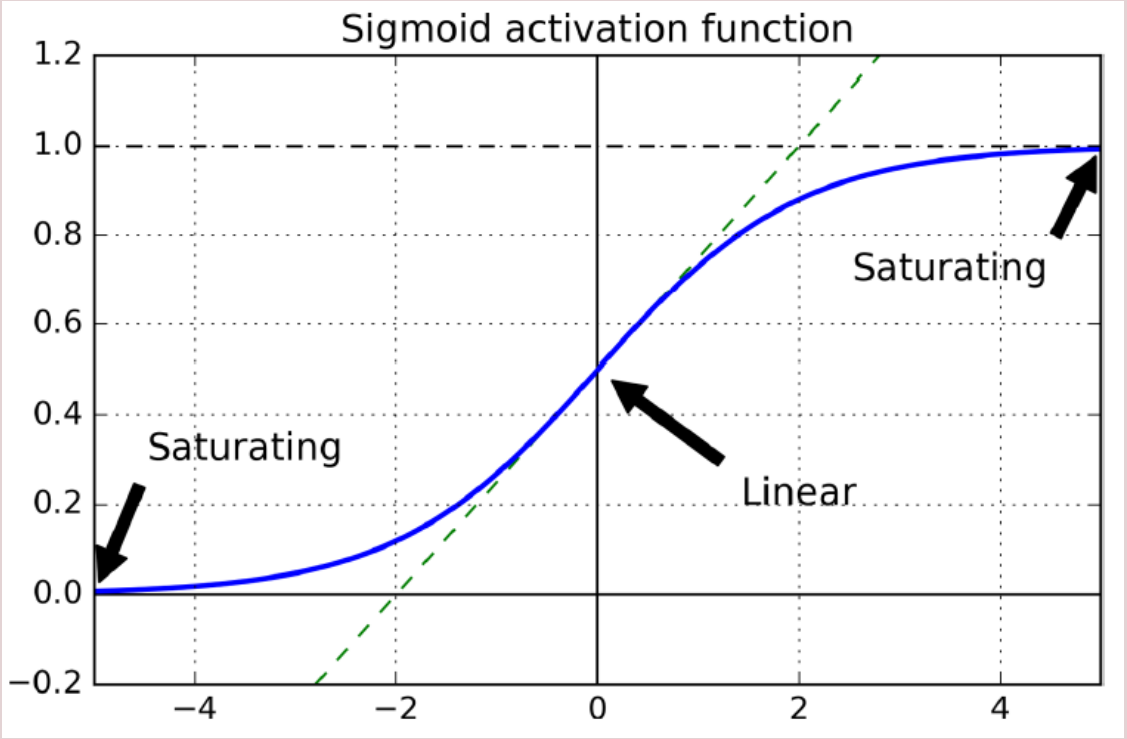


Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Weaknesses: Vanishing and Exploding Gradients

- **Vanishing gradients problem**: Gradients get smaller as the layers get deeper, which means the deeper layers are almost never updated and the training does not converge.
  - **Why:** *Saturating non-linear activation function (e.g., sigmoid) and weights initialization (e.g., zero-mean unit-variance Gaussian).*
- **Solution:** Use ReLU-based activation functions and better initialization schemes.

makin dalem gradient makin vanish

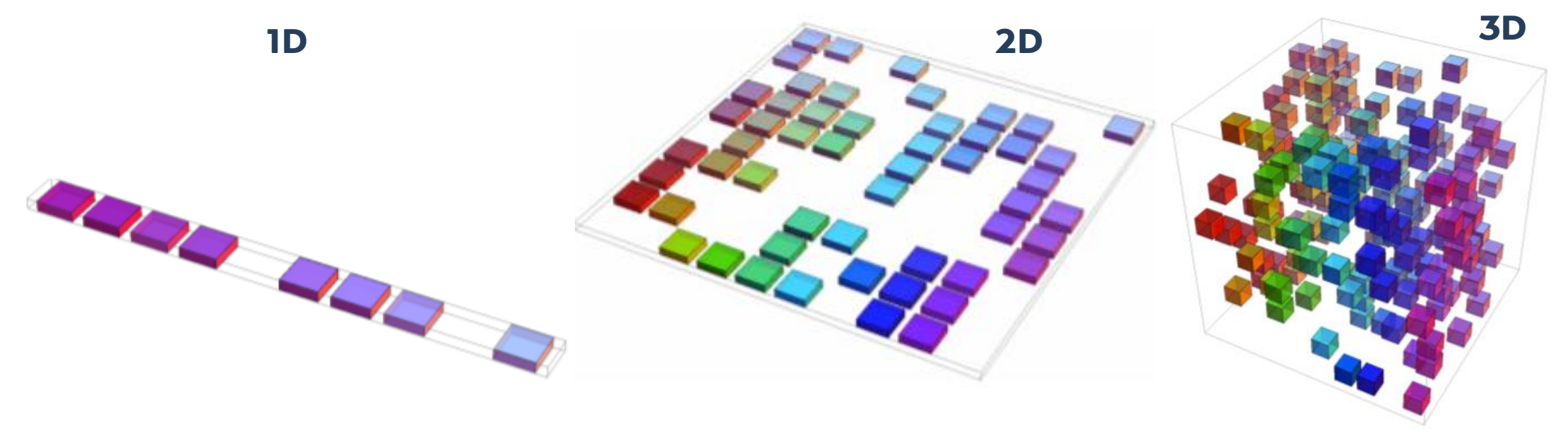


Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Weaknesses: Curse of Dimensionality

more gak selalu better. Sama seperti neuron/layer, lebih banyak blm tentu lebih baik.

- Many machine learning problems are very difficult when the number of dimensions is large because higher dimensions need more data. With less data, the ML model is more prone to overfitting.
- **Solution:** Use data augmentation and/or transfer learning and regularization schemes to avoid overfitting.



Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Weaknesses: Overfitting and Underfitting

- Example: Polynomial regression is easy to overfit and underfit (left figure).
  - Let's say we have a data set that follows quadratic form.
    - The high-degree polynomial regression model (i.e., $degree = 300$ (green)) is severely overfitting the training data, while the linear model (i.e., $degree = 1$ (red)) is underfitting it.
- There is also a term called "**the bias & variance trade-off**".
  - High bias (underfitting): The ANN is too small (simple).
  - High variance (overfitting): The ANN are too big (complex).

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

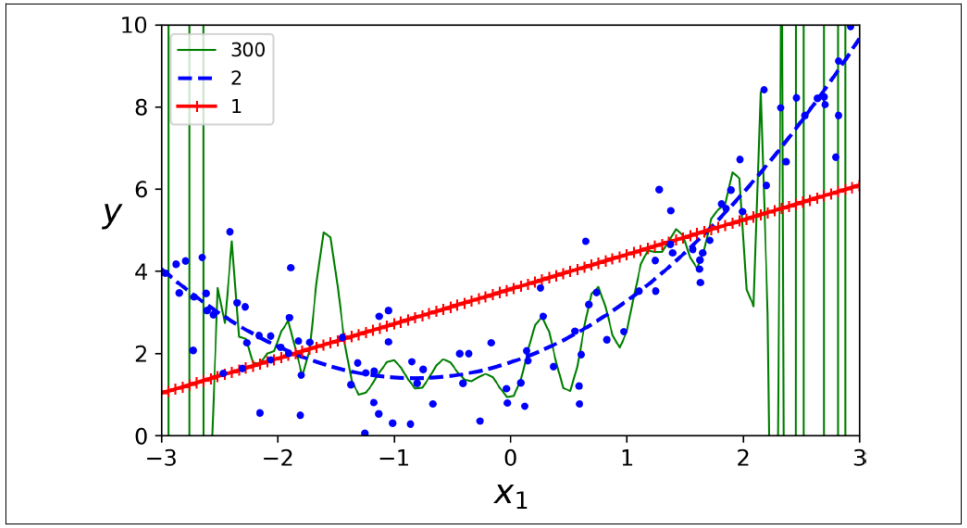
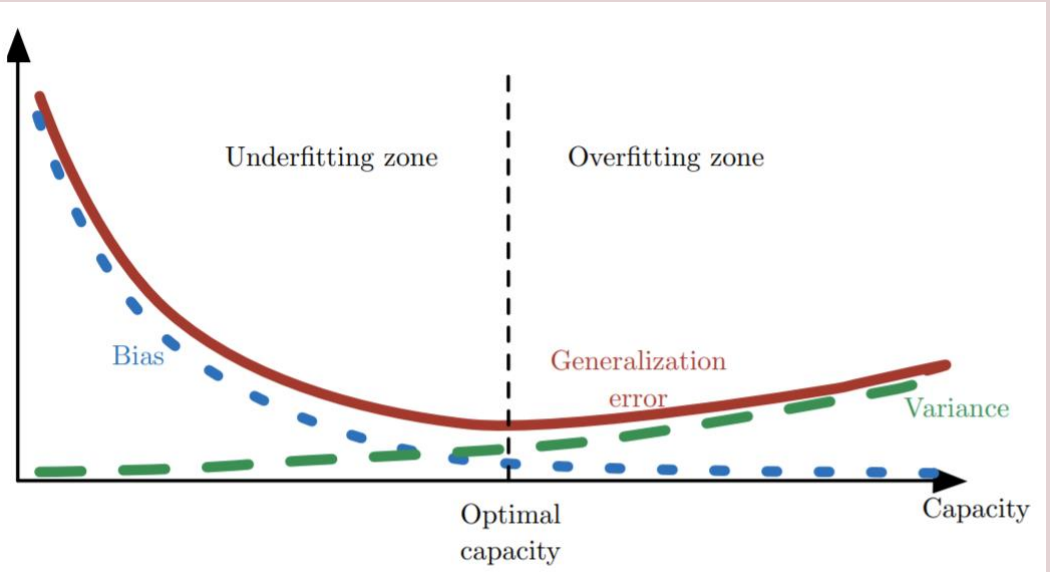


Figure 4-14. High-degree Polynomial Regression



Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Weaknesses: High Numbers of Parameter

- *Fully connected layer* (or usually called *dense layer* in deep learning).
  - Needs full connectivity between input and output layer
  - Results in a large *matrix multiplication*.
- Results in ineffective computation
  - Visualization

kalau setiap layer dimasukkan ke network, dengan layer yang besar.

Ini bakal gede banget karena gambar sendiri kayak (1000 x 2000 pixels).

Dulu setiap gambar harus dibuah 30x30 karena mau gak lebih dari 900 pixel yang di proses.

- Solution: Use convolutional layer instead of fully connected layer.

State of the Art

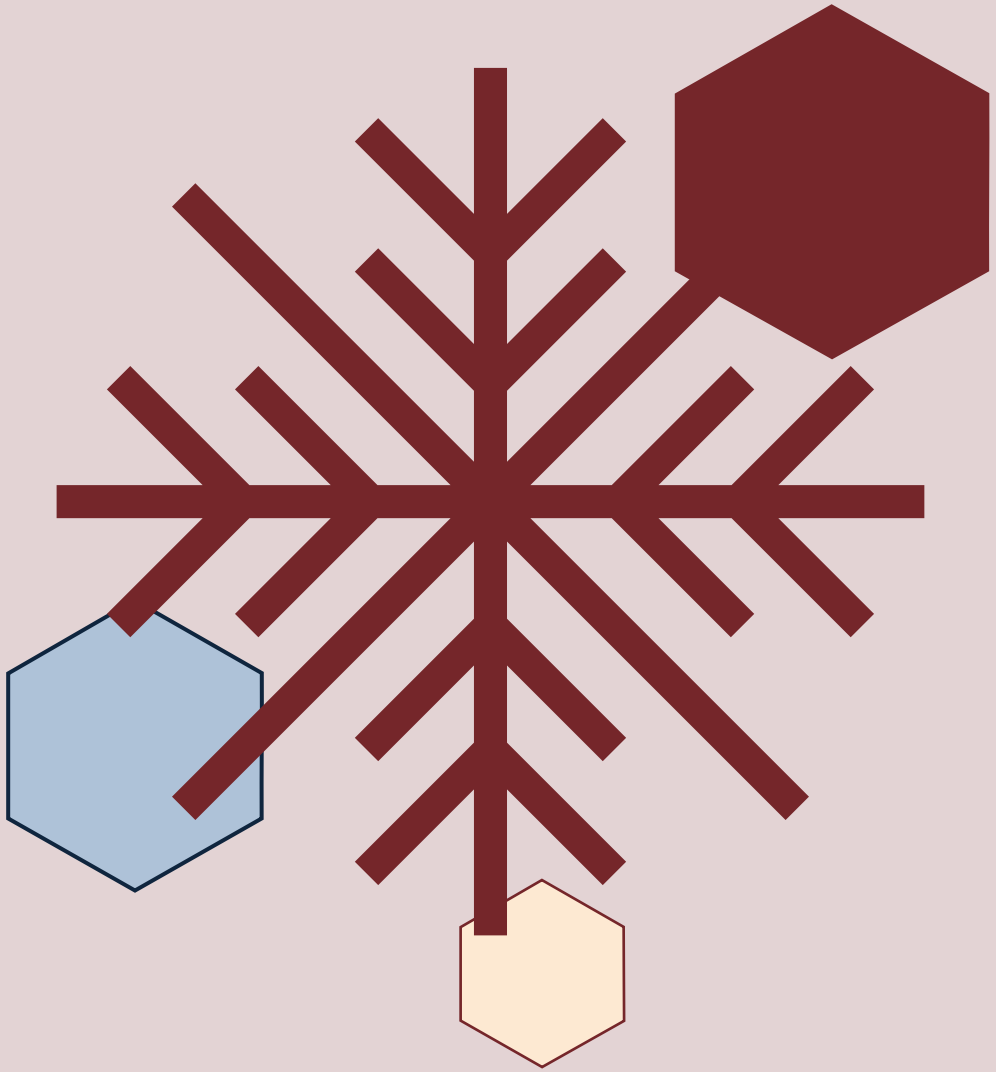
dari weakness itu, muncul namnya convolution.

Ketika kita membicarakan connected layer, yang canggih itu sekarang convolution.

- Artificial neural networks have many weaknesses, which made their popularity went down in the 2000s.
- CNN and deep learning can address these weaknesses.
  - Deep learning doesn't exactly solve these issues, but it handles them better so that the networks can still be effective.
- There have been many advances due to high computation power and the availability of data

ML itu gak kasih answer, melainkan prediksi.

Machine learning and AI **do not give certainty** of identification, they give the **statistical probabilities** that can help the human expert understand and decide between the likelihood of different probabilities. (Popovic, et al, 2020)

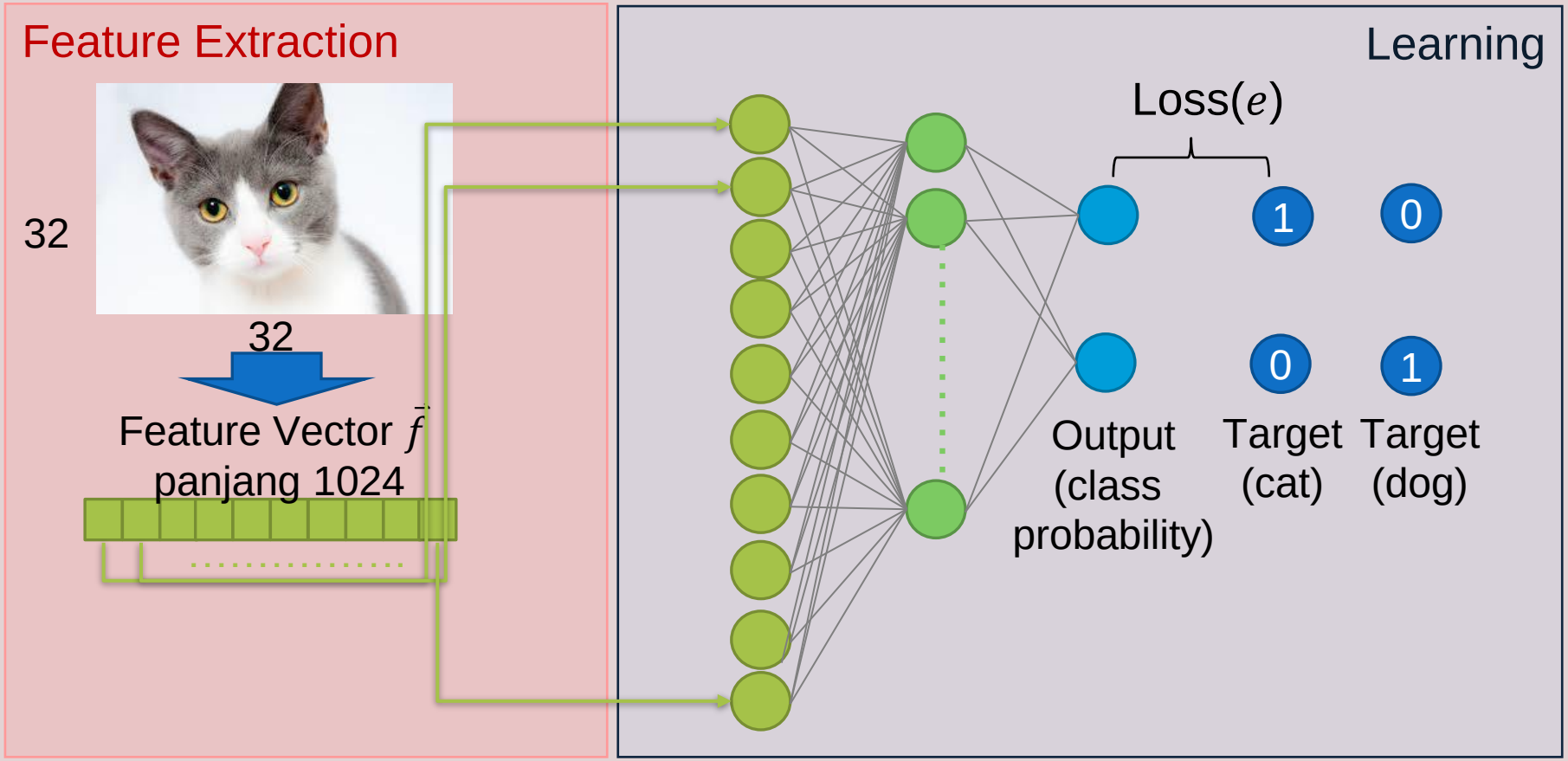


Convolutional Neural Networks

Section 3

Recall: A Simple ANN for Image Classification

dengan menambah konvolusi kita berusaha untuk encode yang ada digambar, supaya besarnya tidak sebesar jumlah pixel.



CNNs perform both feature extraction and learning in the same network.

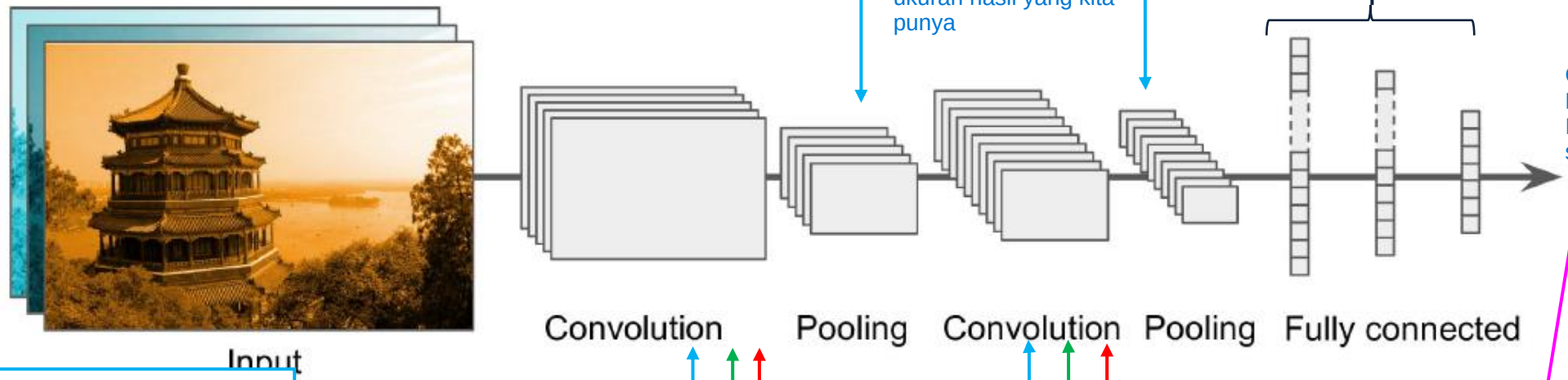
Typical Schematic of Deep Neural Networks: Convolutional Neural Network (CNN)

skema standar CNN

Use **pooling layer** to compress or reduce information.

pooling: mengecilkan ukuran hasil yang kita punya

The fully-connected layer is still useful and used at the last layers.



CNN gak harus ada Pooling/Fully Connected. Ini hanyalah hal yang standard

Use **convolution layers** instead of fully-connected layers.

Use non-saturating non-linear activation function (e.g., **ReLU**).

Use **Batch Normalization** before/after activation funct.

Use modern optimization methods (e.g., **Adam**).

Multiple layers **can** be involved: Convolution layers, Fully connected layers, Pooling layers, Activation layers, *Deconvolution layers*

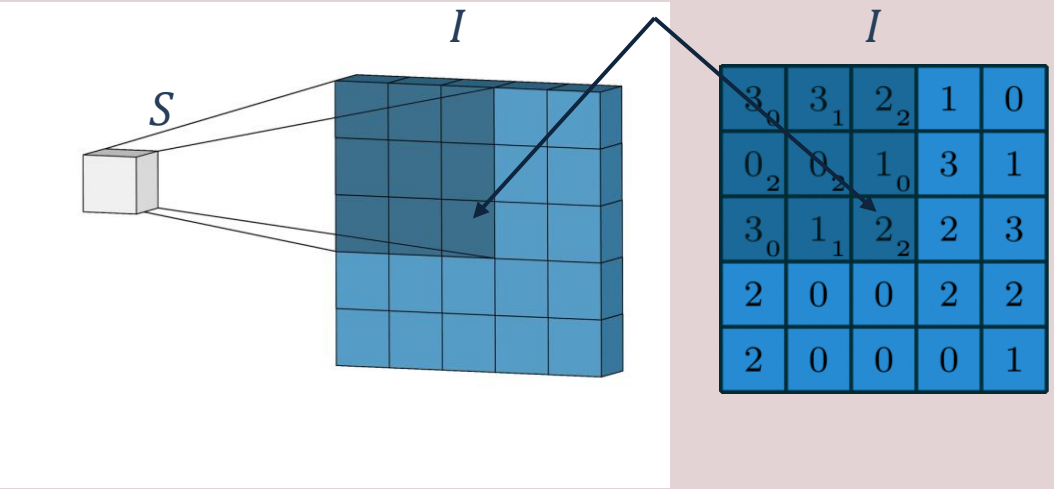
Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

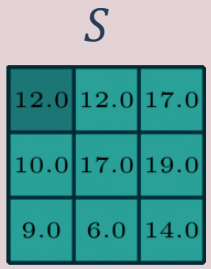
Convolutional Layers

- Convolutional layer is the most important building block of a CNN.
- ***Convolution*** adalah sebuah operasi matematika yang menggeser satu fungsi ke fungsi lainnya dan mengukur integral dari *pointwise multiplication* keduanya.
- In most libraries, convolutional layers implemented cross-correlations, which is very similar to convolution.
- **Cross-correlation function** between two-dimensional image I with size (i, j) and kernel filter K with size (m, n) where S is the resulted image.

ada filtering, dimana menerjemahkan gambar.

Konvolusi melihat: ada pola apa digambar (ada kayak garis apa, warna apa, dll)





$$S(i,j) = (K * I) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$

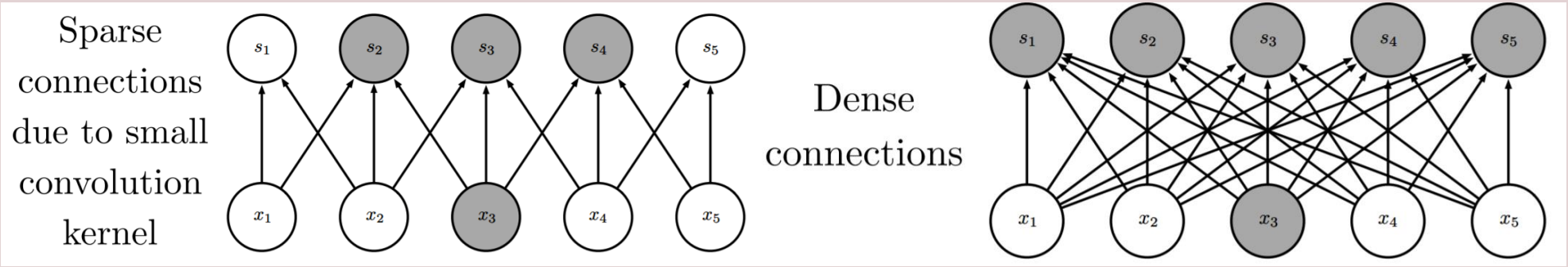
(Illustrations by Irhum Shafkat at https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1)

Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Three Important Attributes of Conv. Layer

apabila kita pakai conv layer, kita mengurangi connectivity

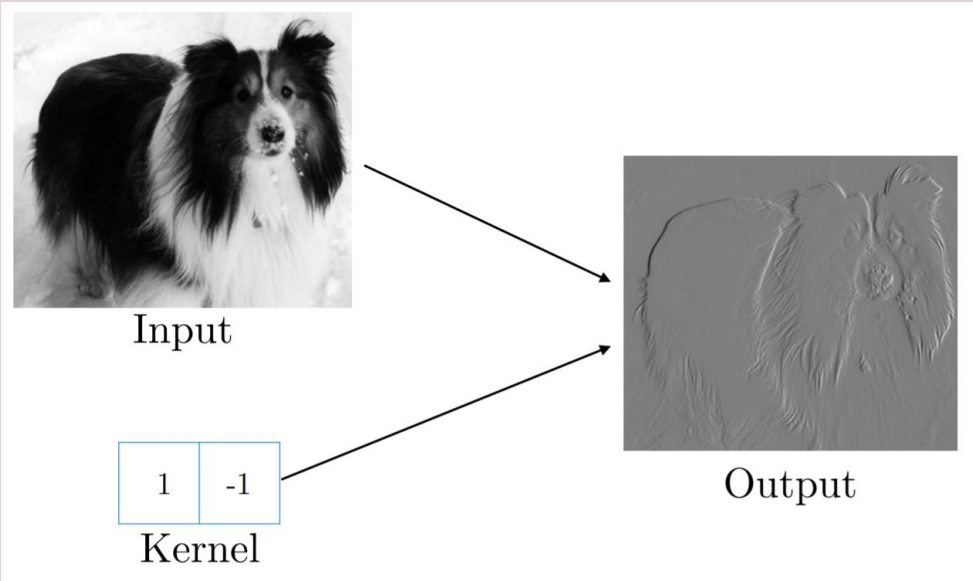
1. Sparse connectivity
   - Limited connections between output units and input units on the local space → vs basic neural networks, where all input units are connected to every output unit resulting in a very large matrix multiplication
   - The advantages of sparse connectivity are fewer parameters, better statistical efficiency, a focus on local, small, and meaningful features, and lower computational costs.



Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Three Important Attributes of Conv. Layer

2. Parameter sharing
   - Sharing parameters for more than 1 input unit: the connection between input and output units is shared with other input-output unit pairs
   - Shows that a group of weights (feature detector) can be used in multiple locations of the image



Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Three Important Attributes of Conv. Layer

3. Equivariant representation to translation
   - Mempunyai arti *convolution* menghasilkan hasil yang sama apabila gambar diubah oleh *shift operation* (i.e., *translation*).
   - Ini menyiratkan bahwa suatu fitur masih dapat dideteksi meskipun dipindahkan ke lokasi lain.
   - Namun, perlu dicatat bahwa: *convolution is naturally not equivariant to other transformation such as rotation and scaling*.

kalau ada bentuk anjing diatas vs dibawah

maka bentuk representasi vektor akan tetap sama.

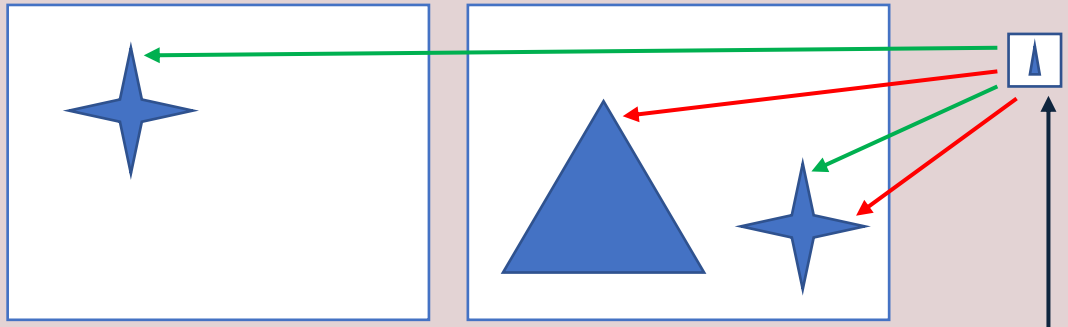


Image #1 *Image #2*

Kernel di sebelah kiri dapat mendeteksi ujung bintang yang menghadap ke atas dimanapun bintang berada (*green arrows*). Namun, kernel yang sama tidak dapat mendeteksi ujung bintang yang telah dirotasi (*rotation*) dan lebih besar/kecil (*scale*) (*red arrows*).

Zero Padding and Stride of Convolution Layer

- **Zero padding:** Adding zeros around the image input to ensure output has the same size as input karena tanpa padding, kita tidak dapat melakukan filtering pada ujung2 piksel
- **Stride:** Connecting a larger input layer with a smaller hidden layer after it by skipping over pixels on the receptive field. Default $\text{stride} = 1$. kita skip beberapa aja, biar gak di detect semua.

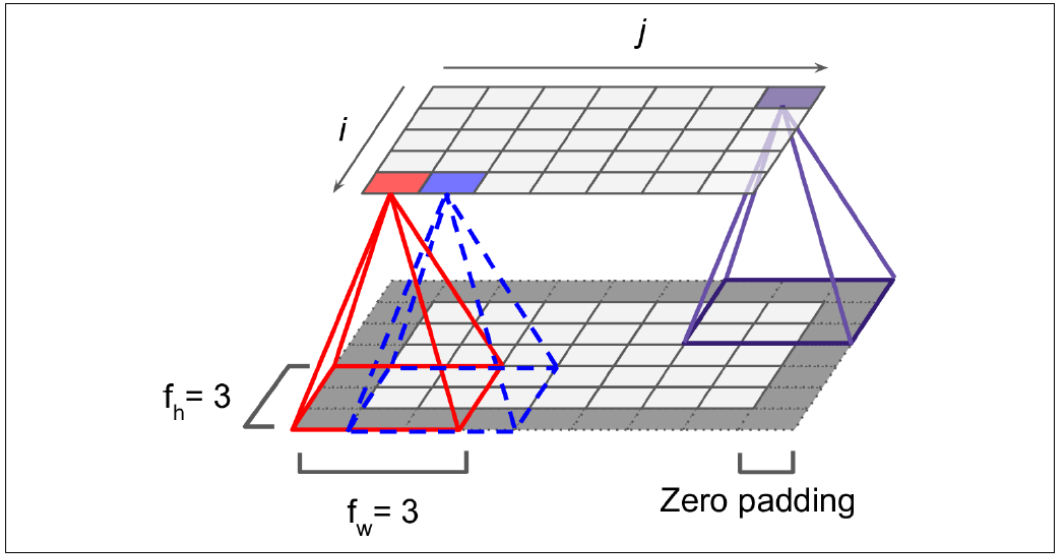


Figure 14-3. Connections between layers and zero padding

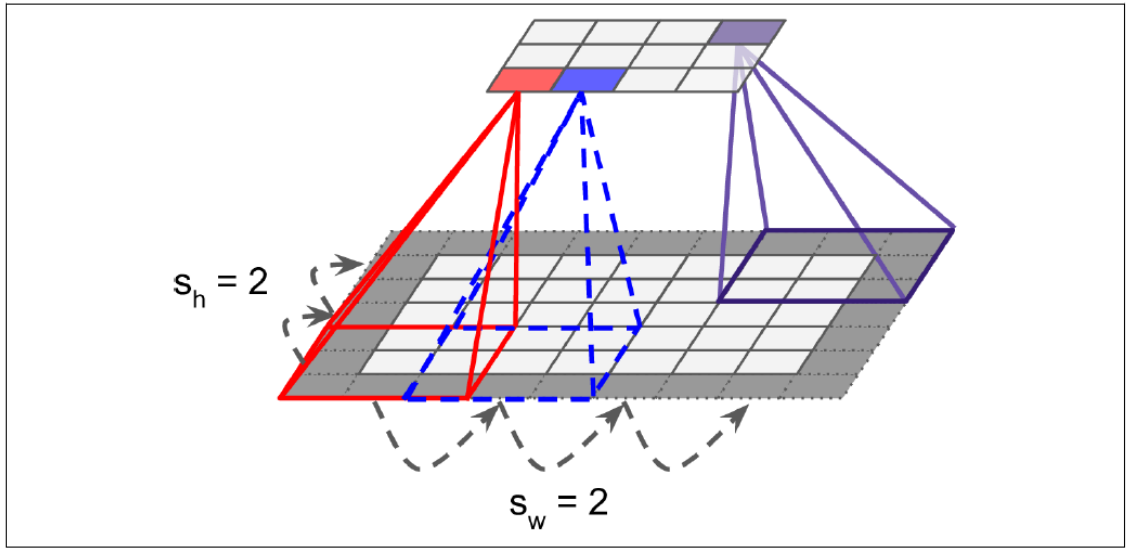


Figure 14-4. Reducing dimensionality using a stride of 2

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Feature Maps from Multiple Convolution Kernels in a Layer

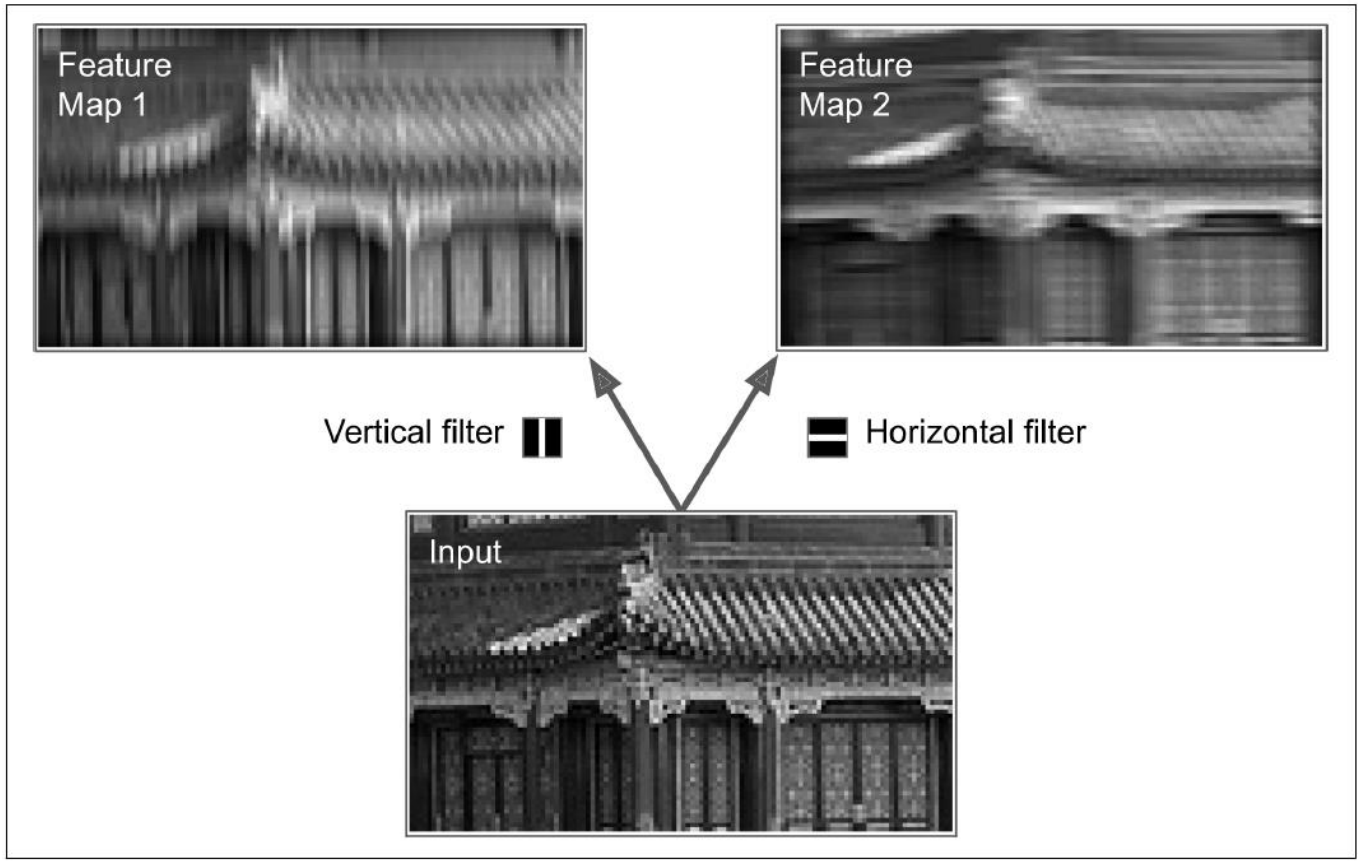


Figure 14-5. Applying two different filters to get two feature maps

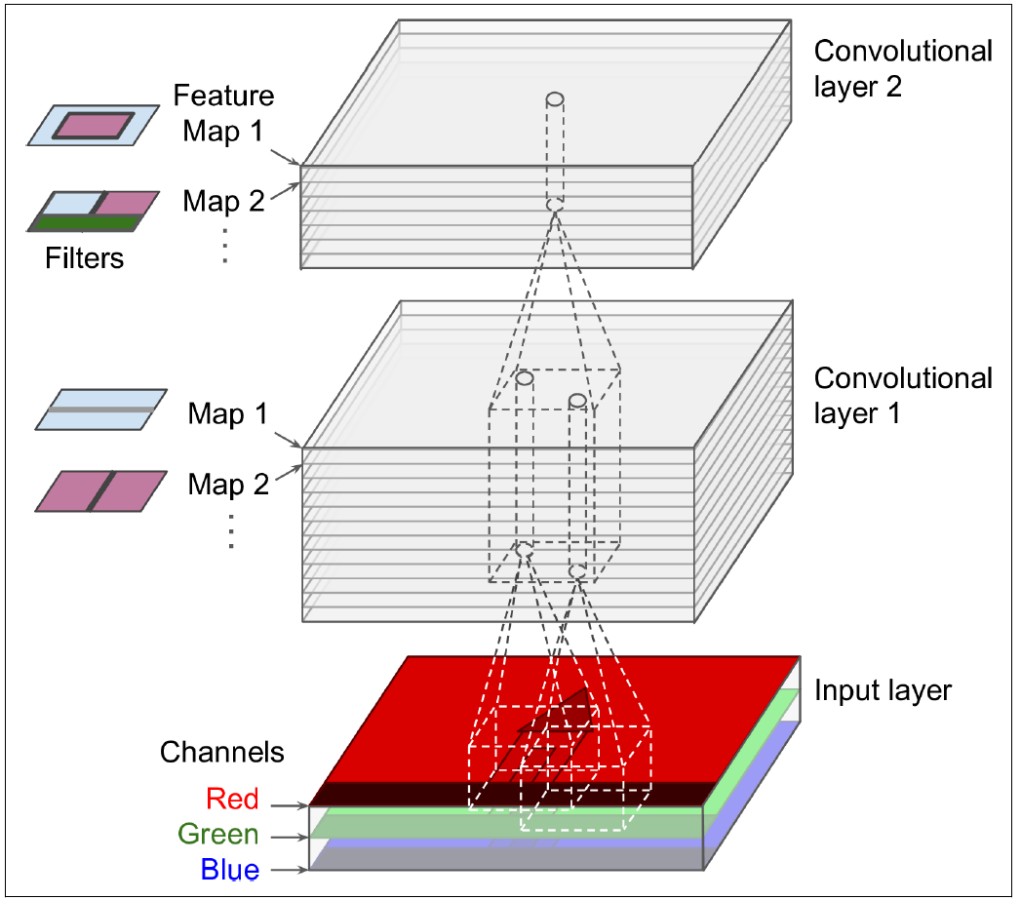
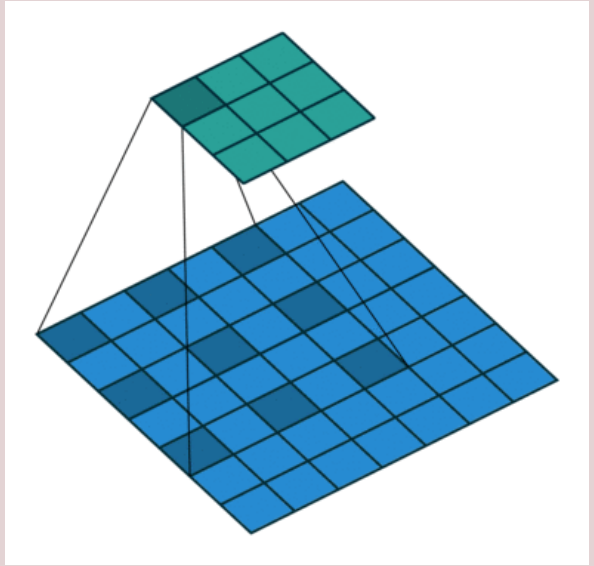


Figure 14-6. Convolution layers with multiple feature maps, and images with three color channels

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Variations: Dilated Convolution

- Kernels are distances with a dilation factor (l). With a dilation factor > 1 (i.e., l>1), we can enlarge the receptive field
- It has been reported to work well in computer vision, but naïve application of dilated convolutions does not always improve performance (Hamaguchi et al., 2018).



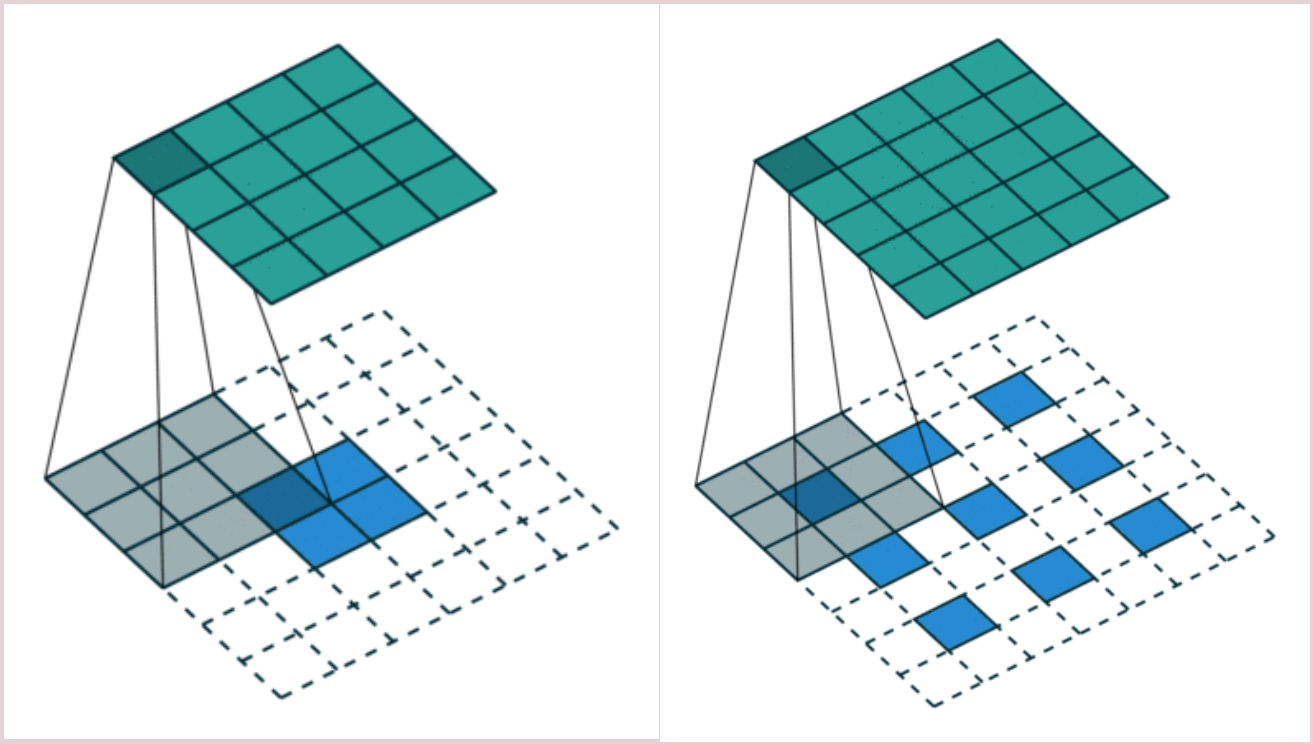
Dilated Convolution (l=2). Illustration is from https://towardsdatascience.com/review-dilated-convolution-semantic-segmentation-9d5a5bd768f5

Deconvolutional Layer

data yang dipunya 1000x1000
vs

data yang dipunya 500x500 beda dan terkadang kita mau liat data pada representasi yang lebih besar.

- A bit misleading: a deconvolution layer performs also convolution
- Transposed / Fractionally-strided / Inverse convolutional layer



https://datascience.stackexchange.com/questions/6107/what-are-deconvolutional-layers
Shi, W., et al (2016). Is the deconvolution layer the same as a convolutional layer?. *arXiv preprint arXiv:1609.07009*.

Pooling Layers

- Subsampling the input image to reduce computational requirements, memory usage, and number of parameters (reducing the risk of overfitting) (Géron, 2019).
- E.g.,: *max pooling layer*, average pooling layer

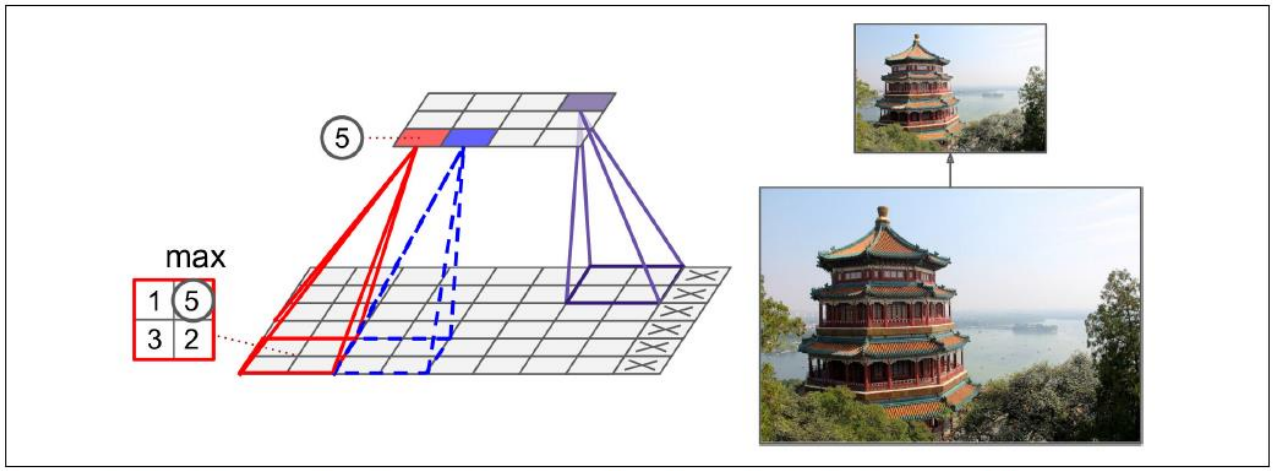


Figure 14-8. Max pooling layer (2 × 2 pooling kernel, stride 2, no padding)

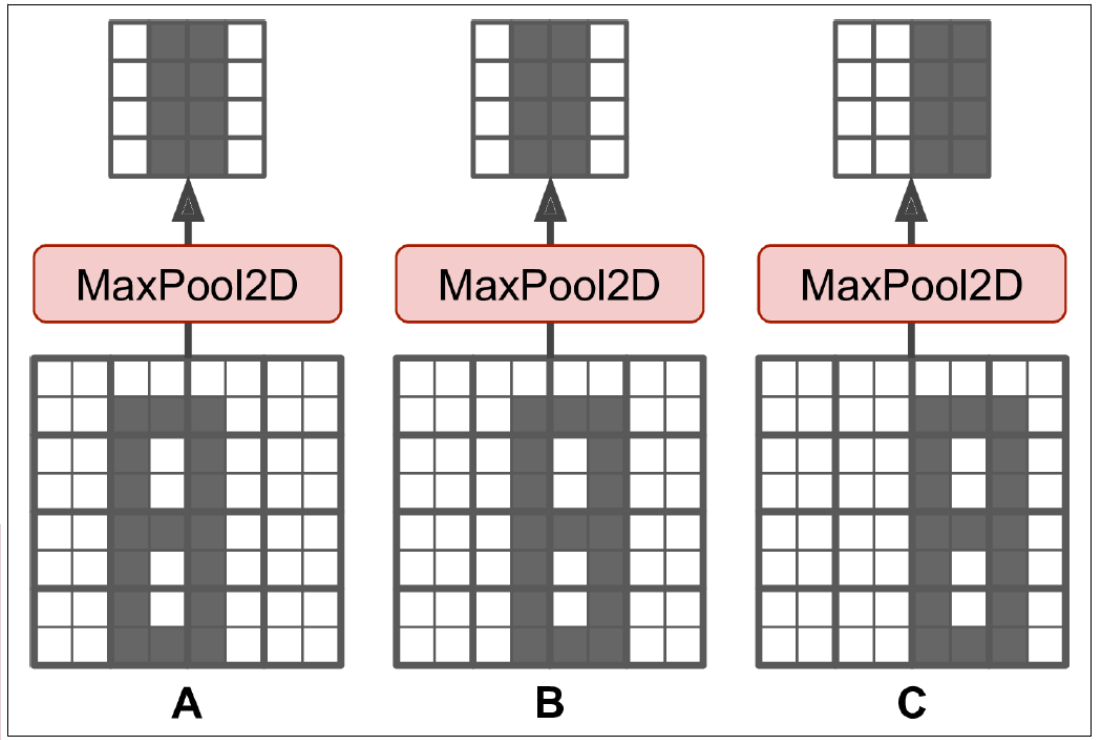
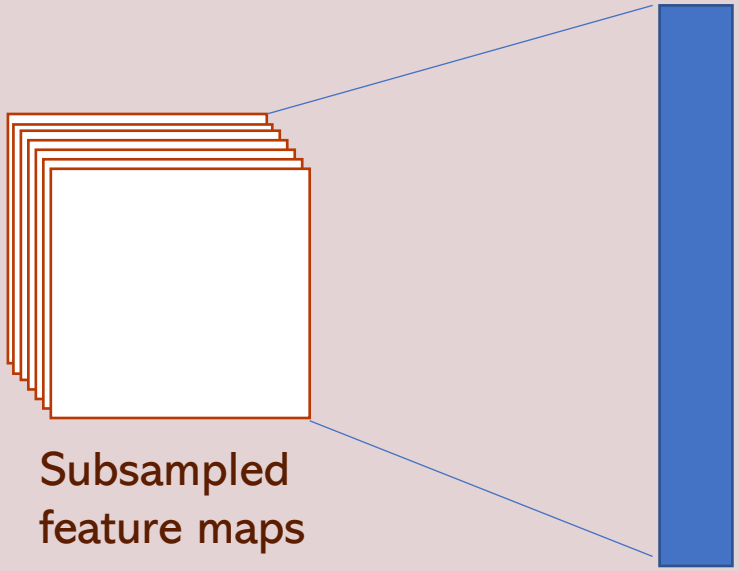


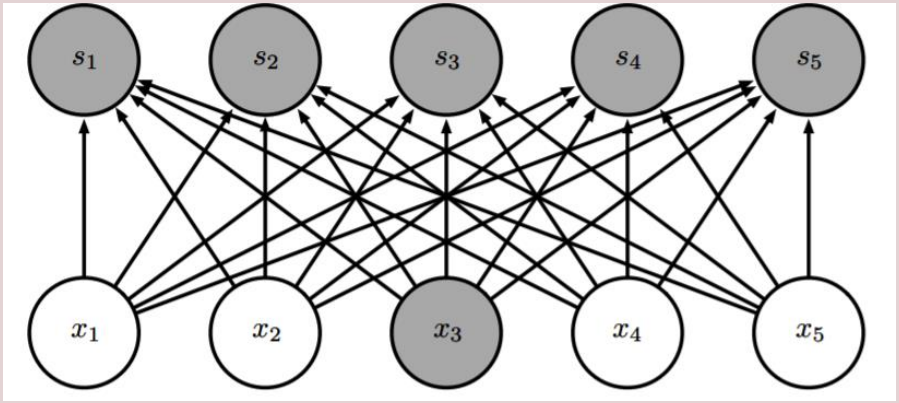
Figure 14-9. Invariance to small translations

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Fully Connected Layers

- Fully connected **from every point** in the feature map **to every point** in the connected layer
- Usually at the end of the network for decision making





Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

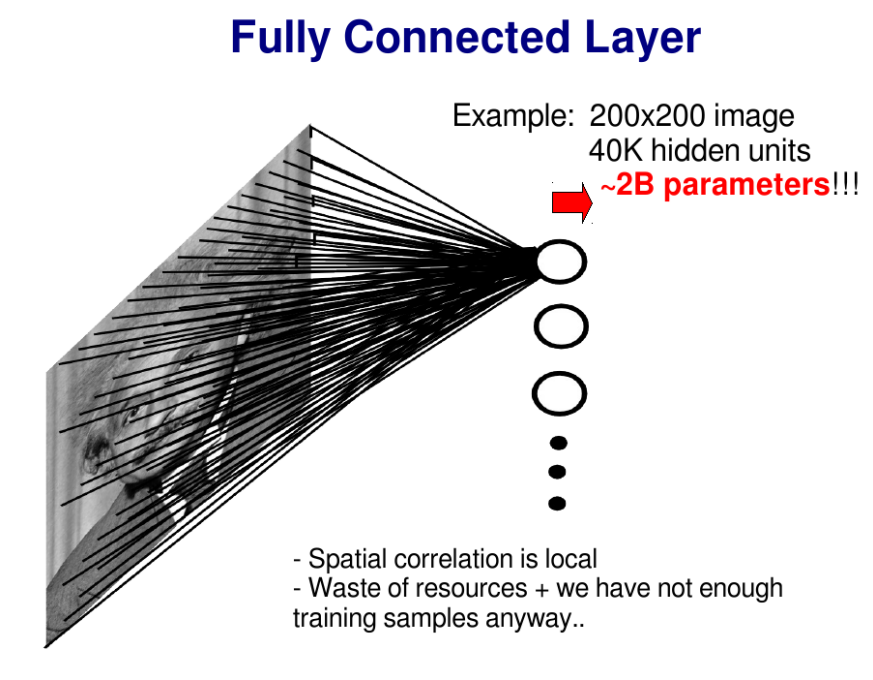
Fully Connected Layer vs. Convolution Layer

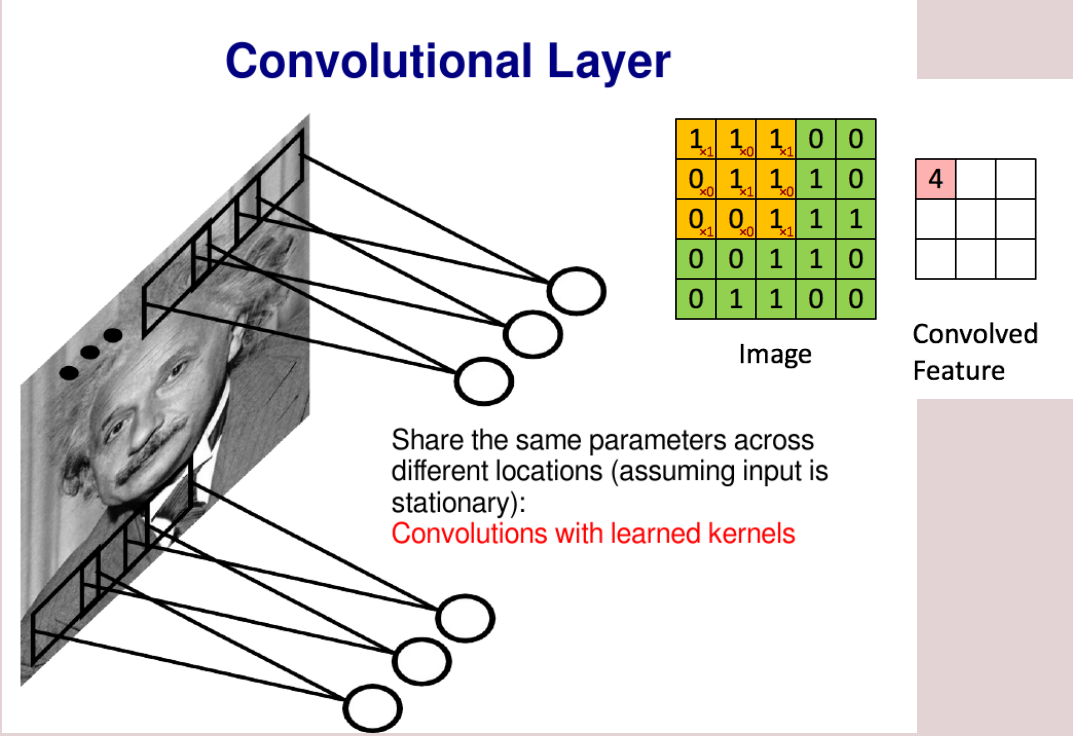
✗ Fully connected networks are not suitable for image processing:

- Too many parameters

✓ Convolutional layers:

- Image content is often stationary (the same or similar content can appear anywhere in an image)





Ranzato, Facebook, “An introduction to deep learning”.

Sumit Saha, towardsdatascience.com, A comprehensive guide to convolutional neural networks. Edited by Charissa Poon.

Non-Saturating Activation Functions [Additional]

- Non-saturating activation functions are used to tackle vanishing gradient problems.
- Some common ones:
  - **Rectified Linear Unit (ReLU)**
  - Leaky ReLU
  - Parametric Leaky ReLU (PReLU)
  - Exponential Linear Unit (ELU)
- Some notes about sigmoid function:
  - In neuroscience, biological neurons have similar activation functions to sigmoid (S-shaped), so in the beginning stages of deep learning, this function was used for a long time
  - Recently, studies show that, recent studies show that ReLU work better for ANNs. This is an example where biological analysis was misleading.

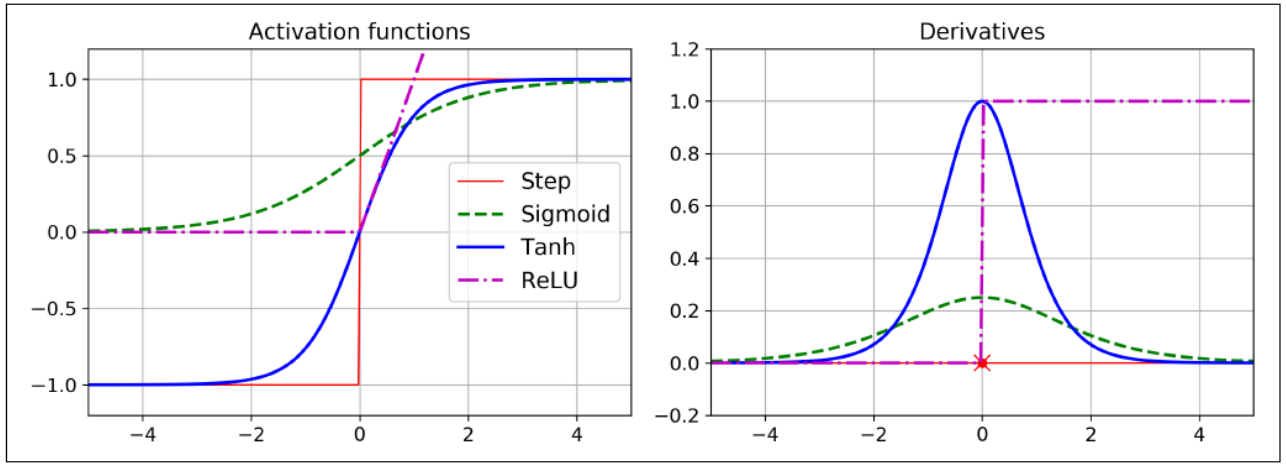


Figure 10-8. Activation functions and their derivatives

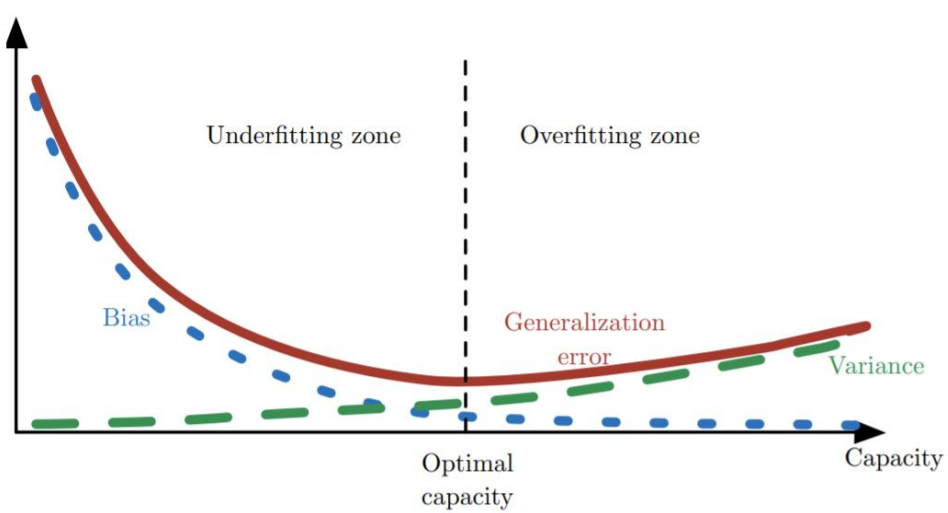
Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Batch Normalization [Additional]

- Better weights initialization and non-saturating activation function can reduce the risk vanishing/exploding gradients significantly in the beginning of training, but it doesn't ensure that the same problems will not arise later
- **Batch Normalization (BN)** was proposed to handle vanishing/exploding gradients (Loffe, S. and Szegedy, C., 2015)
  - BN adds an operation in the model right before or after the **activation function on each hidden layer for** zero-centering and normalization on each input, then scaling and shifting on the feature maps.
  - BN "normalizes" the resulting feature map before inputted into the next convolutional layer
- In many cases, if we add a BN layer as the very first layer of our neural network, we do not need to standardize your training set: the BN layer will do it for us.

Regularization [Additional]

- **Regularization** are the modifications we make on the learning algorithm to reduce validation error, not training error.
- Often used: L1 norm and L2 norm; Elastic Net (i.e., L1 + L2 Norm); Early stop
- Note the Bias & Variance trade off: Deep neural networks typically have tens of thousands / millions of parameters. With so many parameters, the network has an incredible amount of variance.



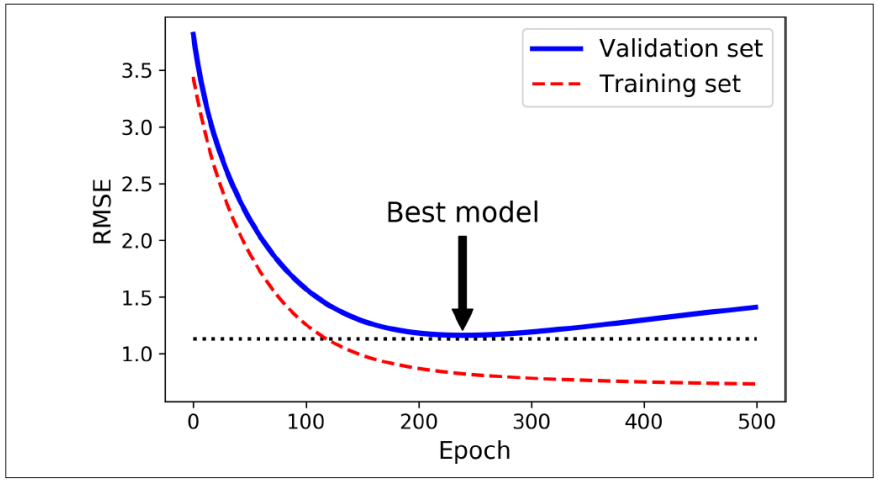
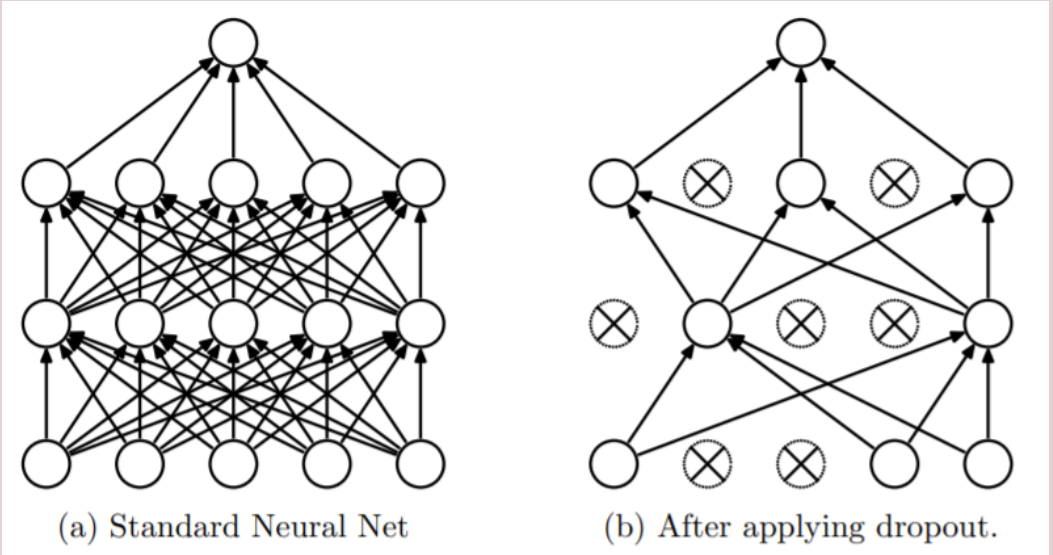


Figure 4-20. Early stopping regularization

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.
Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Dropout Layer [Additional]

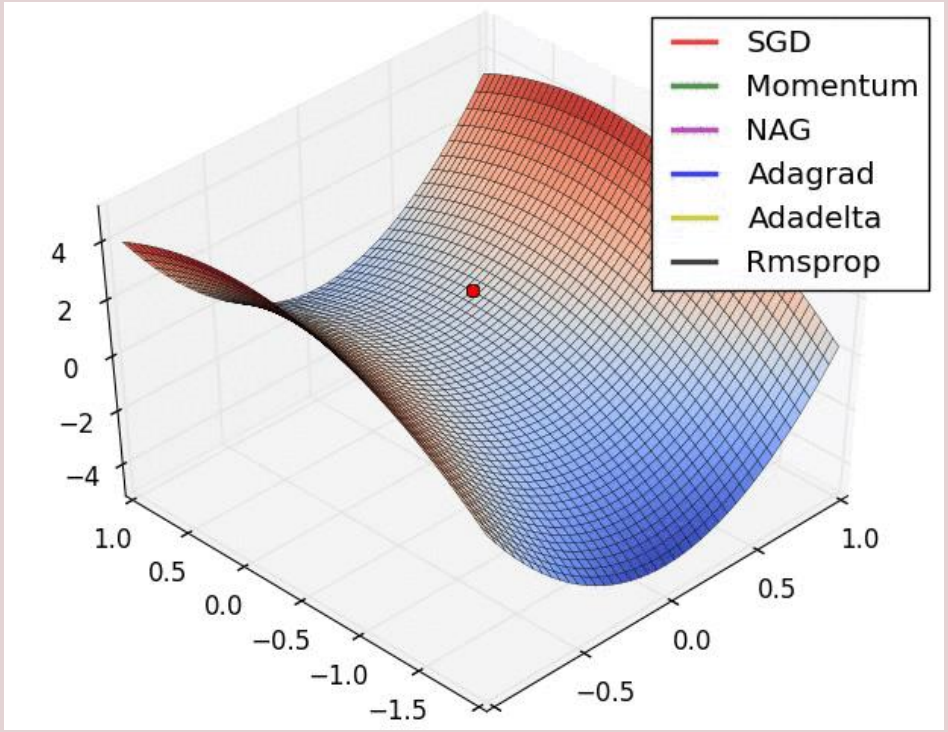
- ***Dropout*** is one of the most popular regularization techniques for deep neural networks.
  - Note: Regularization is a method used to avoid overfitting.
- ***Dropout:*** *At every training step, every neuron (including the input neurons but always excluding the output neurons) has a probability p of being temporarily "dropped out," meaning it will be entirely ignored during this training step but may be active during the next step. After training, neurons do not get dropped anymore.*
- The hyperparameter p is called the dropout rate, and it is typically set between 10-50%:
  - Closer to 40-50% in convolutional neural nets.



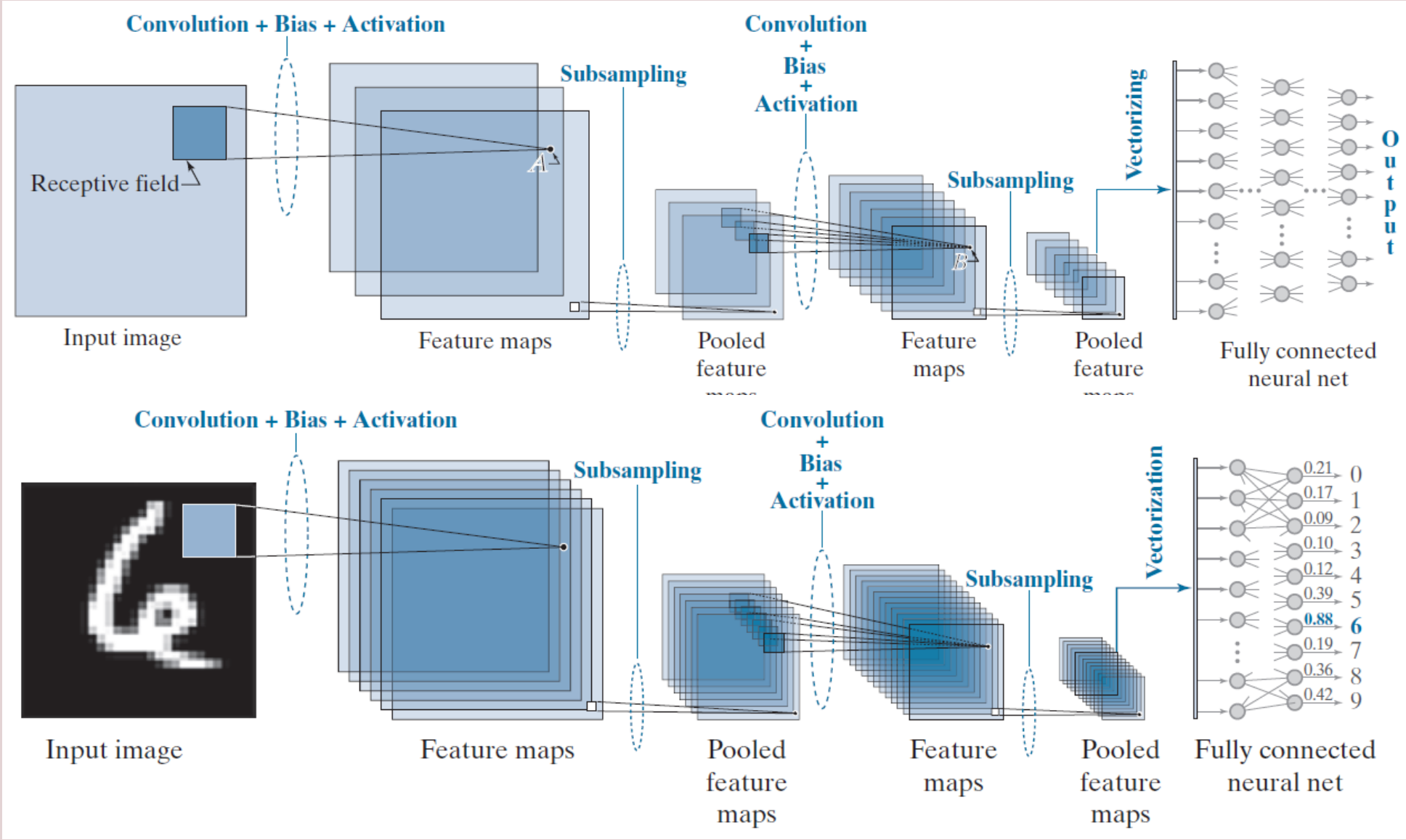
Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.

Better Optimization than SGD [Additional]

- The most commonly used optimization for CNN is Adam (*adaptive moment estimation*).
- Adam combines the ideas of momentum optimization (i.e., *exponentially decaying average of past gradients*) and RMSProp (i.e., *exponentially decaying average of past squared gradients*).
- Generally, Adam is much better than Stochastic Gradient Descent (SGD).



The Very First Working CNN: LeNet

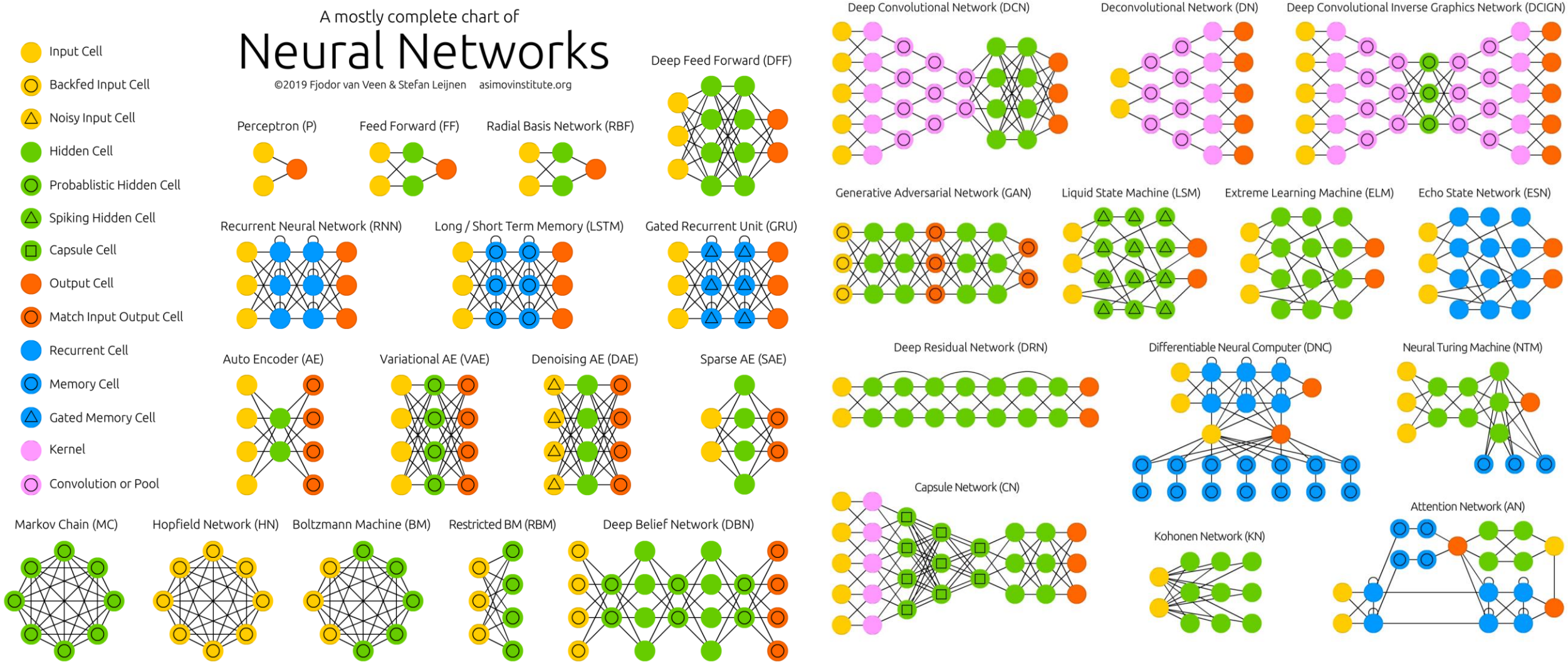


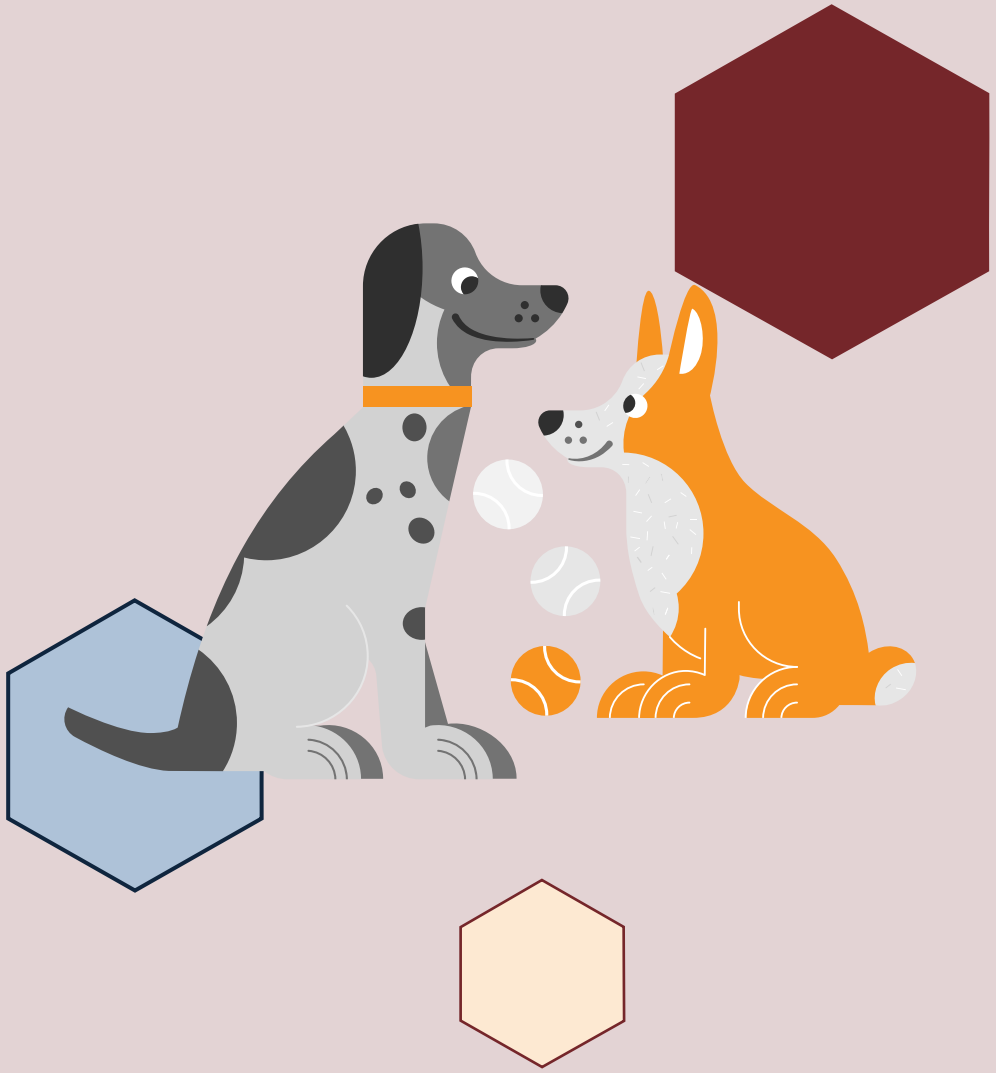
Gonzalez, R. C., and R. E. Woods. "Digital Image Processing, 4th Global Edition."

The Zoo of Neural Networks

convolution melakukan segala hal yang sudah kita pelajari sampai saat ini:

- contohnya edge detection





CNN for Computer Vision Recognition Tasks

Section 4

Image Classification

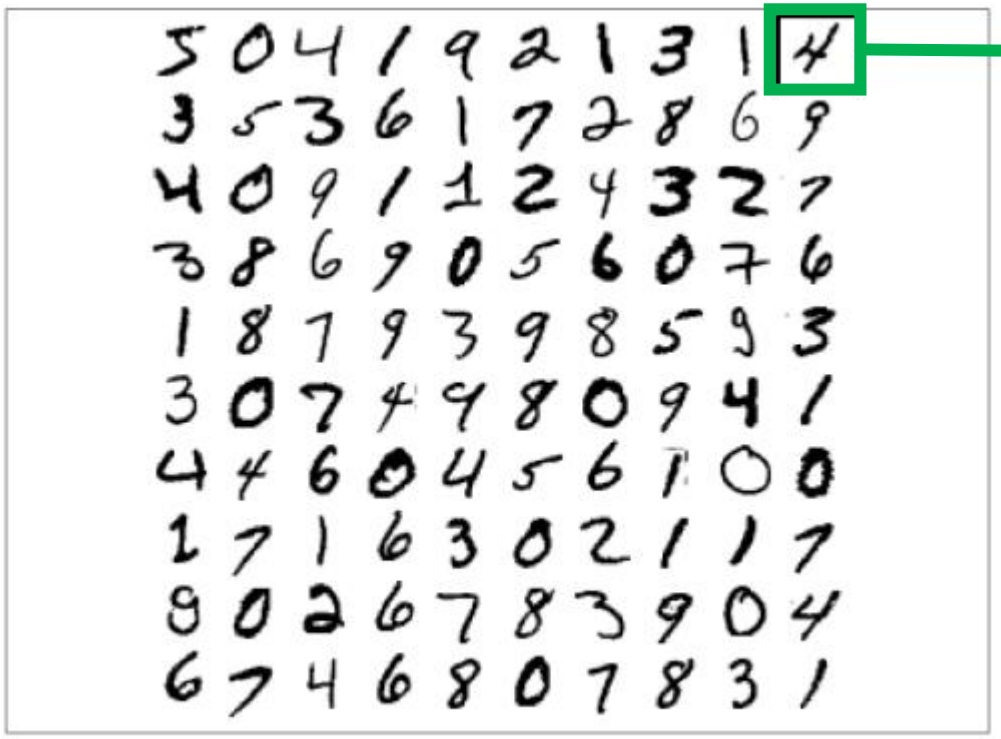


Figure 3-1. A few digits from the MNIST dataset

- When presented with a picture of a written digit – how can the system correctly recognize it?
- When presented with a picture of **any** object – how can the system correctly recognize it?
- Can be evaluated with confusion matrix measurements (e.g., accuracy, precision, recall, etc.), F1-score (Dice similarity coefficient), ROC curve, etc.

Deep Neural Networks for Image Classification

- In this presentation:
  - ✓ LeNet-5
  - ✓ VGGNet
  - ✓ Residual Networks (ResNet)
- Independent reading:
  - ✓ AlexNet
  - ✓ GoogLeNet
  - ✓ Xception
  - ✓ Squeeze-and-Excitation Network (SENet)

LeNet-based Architecture

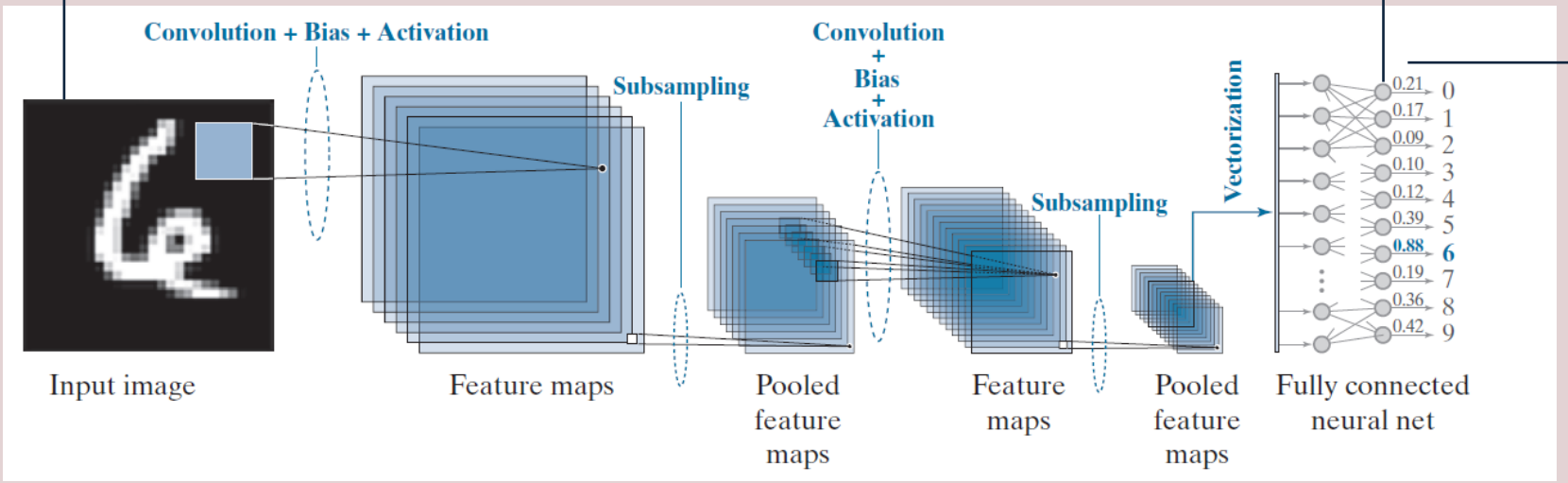
MNIST images are 28x28, but zero-padded into 32x32

Equation 4-19. Softmax score for class k

$$s_k(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\theta}^{(k)}$$

Equation 4-20. Softmax function

$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\sum_{j=1}^{K} \exp\left(s_j(\mathbf{x})\right)}$$



- Uses 5x5 convolution
- Uses *tanh activation function* on each *layer, except classification layer*).
- Demo: http://yann.lecun.com/exdb/lenet/

Gonzalez, R. C., and R. E. Woods. "Digital Image Processing, 4th Global Edition."

VGGNet

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

- Much larger and deeper CNN – with a lot of parameters
- VGGNet is the runner-up of ImageNet Large Scale Visual GoogLeNet.
- VGGNet is commonly used for style transfer, transfer learning, etc.

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

Residual Networks (ResNet)

untuk mengurangi kompleksitas, kita bisa menggunakan residual unit

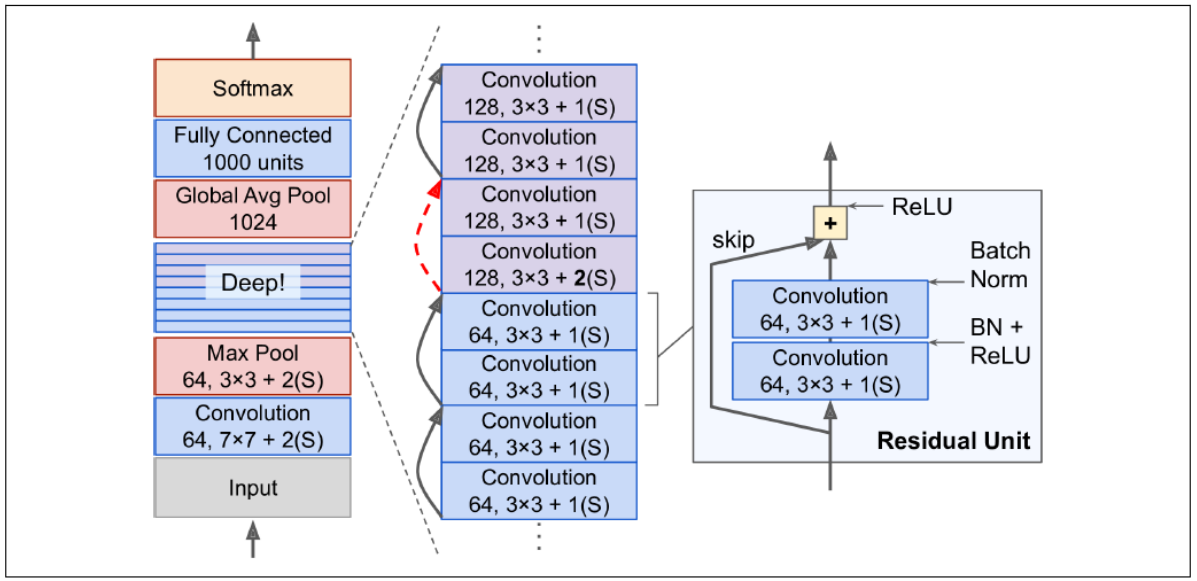


Figure 14-17. ResNet architecture

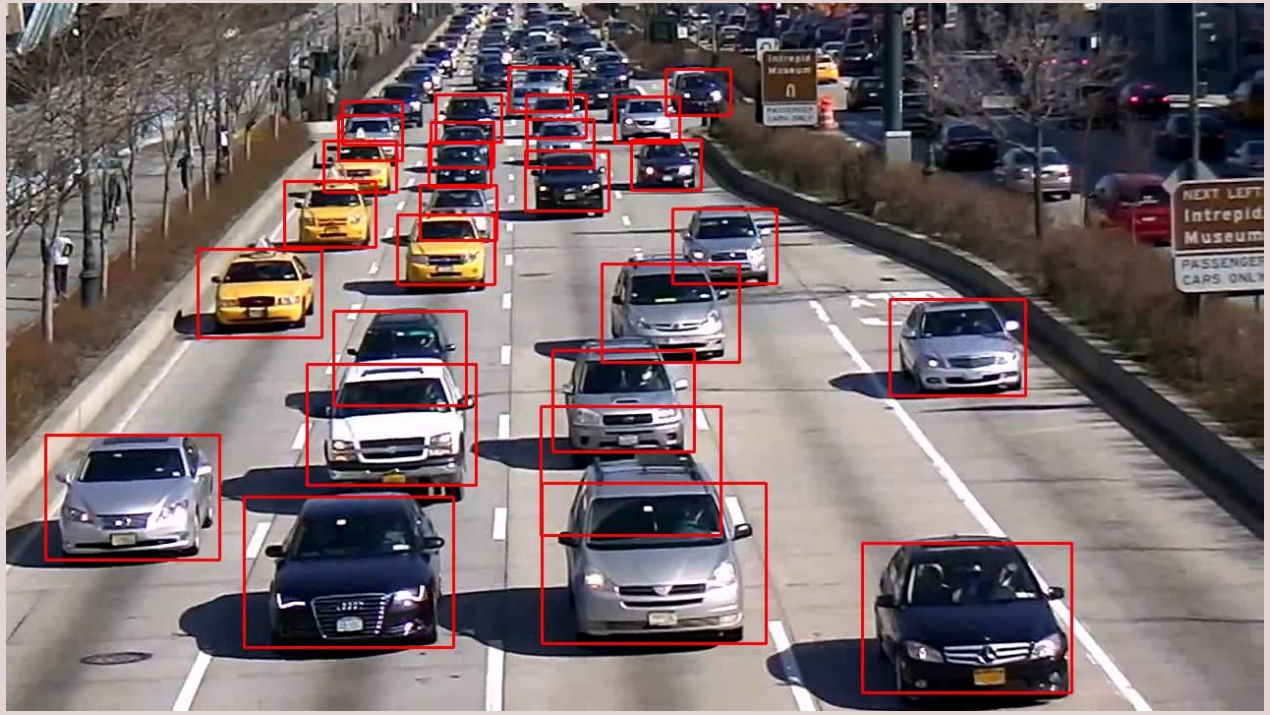
- Uses a proposed **residual unit (RU)** that enables us to use a very deep CNN (152 *trained layers*).
- Every RU has a **skip connection** (*shortcut connection*) to help *forward and backward passes.*
- Note: At the beginning, an RU is very close (or equal) to an identity function (since almost all weights on the convolutional layer are close to zero). This makes the RU easy and fast to train

Semuanya ini didapatkan dari memahami bagaimana MLP dan convolutional layer

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Recall Object Detection using SIFT

- Corner and SIFT features are good for *one-to-one* object detection.
- However, they are not enough for object(s) with high variance (*one-to-many*).
  - Popular example: Car detection



Object Detection

Figure 14-23. Intersection over Union (IoU) Metric for Bounding Boxes

- ***Object detection:*** Not only the class, but also the location (localization).
- Use bounding box as the location: each object has a tuple as feature.
  - (images_id, (class_label, bounding_boxes))
- The "*bounding_boxes*" feature itself has 4 values:
  - (x, y) as the center coordinate of the object
  - $(height, weight)$ as the size of bounding box
- *Deep neural networks* is optimized for object detection by minimizing the MSE (mean square error) cost function of the "*bounding_boxes*".
- The performance is measured by using IoU (Intersection of Union).

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Sliding Window-based Object Detection

- Object detection using DNN is usually performed by sliding a window of detection which will cover the entire image.
  1. Bagi citra/gambar menjadi beberapa bagian (misalkan, 6×8 seperti gambar di samping).
  2. Menjalankan DNN (e.g., CNN) menggunakan sliding window (persegi hitam tebal ukuran 3×3) ke semua bagian gambar.
  3. Gambar dan simpan 1 bounding box untuk setiap objek yang dideteksi oleh CNN (persegi merah).
  4. Jalankan metode ***non-maximum suppression*** untuk mendapatkan 1 bounding box untuk setiap objek yang berhasil dideteksi.
- Objects in an image usually have high variance of sizes. Thus, perform sliding window object detection using with different size of sliding window (e.g., 2×2, 3×3, 4×4).

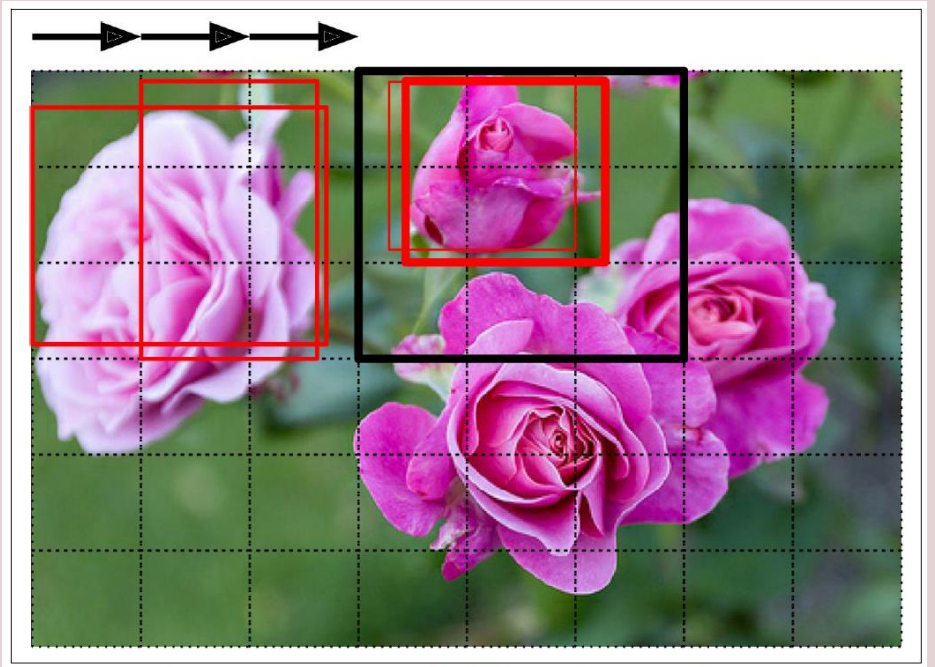


Figure 14-24. Detecting Multiple Objects by Sliding a CNN Across the Image

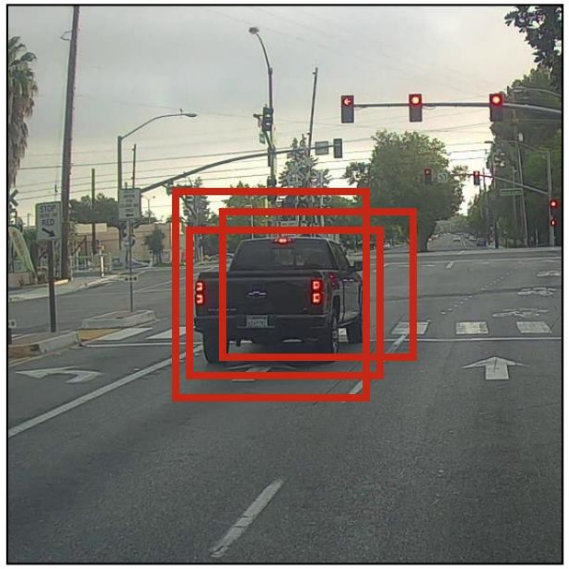
object detection biasa digunakan dengan menggunakan sliding window. Yang dilakukan convolutional filter sama kayak filter di object detection yaitu sliding window.

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Non-Maximum Suppression (NMS)

- The same object could be detected multiple times because of sliding window.
- Use NMS to get the maximum bounding box for each detected object.

Before non-max suppression



Non-Max Suppression

After non-max suppression

Algorithm 1 Non-Max Suppression

```
1:  procedure NMS(B,c)
2:      B_nms ← ∅
3:      for b_i ∈ B do
4:          discard ← False
5:          for b_j ∈ B do
6:              if same(b_i, b_j) > λ_nms then
7:                  if score(c, b_j) > score(c, b_i) then
8:                      discard ← True
9:          if not discard then
10:             B_nms ← B_nms ∪ b_i
11:     return B_nms
```

Non-maximum suppression:
https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c

CNN for Object Detection: YOLO & SSD

- **You Only Look Once (YOLO)** (2016): Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition.
  - Only use 1 module (i.e., CNN) for localization (bounding box) and object classification.
- **Single Shot Multi-box Detector (SSD)** (2016): Liu, Wei, et al. “SSD: Single shot multibox detector." European conference on computer vision. Springer, Cham.
  - Like YOLO, SSD only use 1 module (i.e., CNN) for localization and object detection.
  - Difference between SSD and YOLO: SSD is faster and can detection small objects.
    - SSD could be used in a real-time object detection system (video).
- YOLO and SSD have a lot of derivations
  - YOLOv2, YOLOv3, YOLO9000
  - SSD300, SSD500

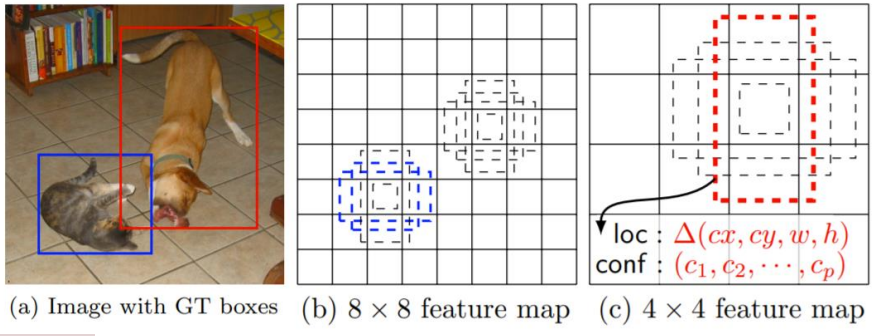
Yolo (you only look once):

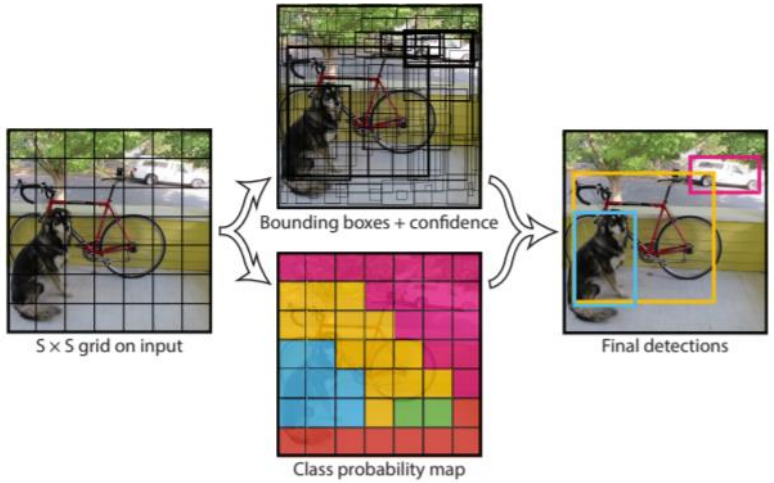
Kalau ada gambar berisi orang

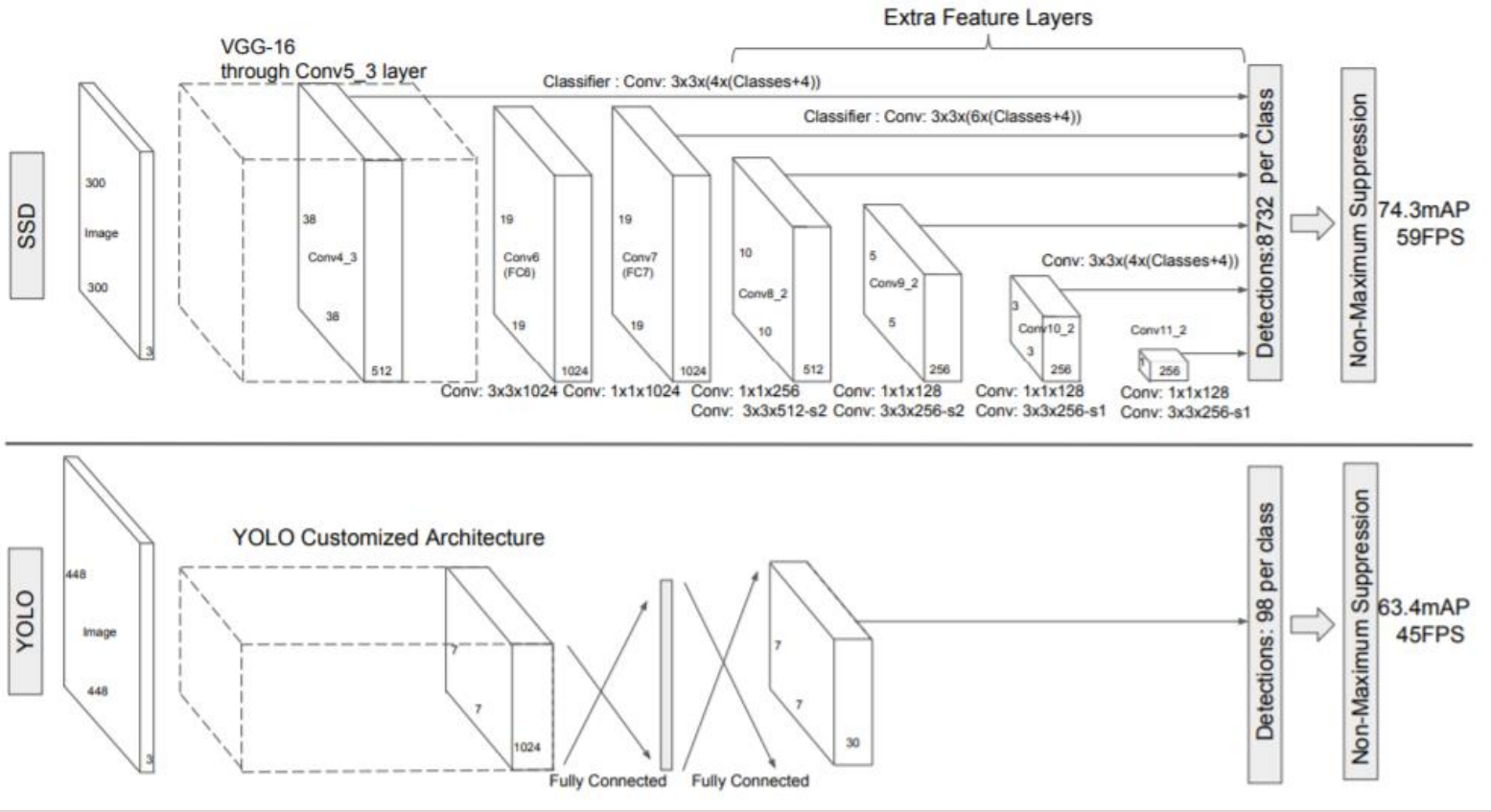
Sebelumnya: kita pakai sliding window, sampai ketemu orangnya, yang kedua lalu hasil deteksi tersebut diklasifikasi jadi orang

Dia langsung tau kalau pakai YOLO

Yolo vs. SSD


(a) Image with GT boxes
(b) 8 × 8 feature map
(c) 4 × 4 feature map
loc : $\Delta(cx, cy, w, h)$
conf : $(c_1, c_2, \cdots, c_p)$


S × S grid on input
Bounding boxes + confidence
Class probability map
Final detections


SSD
300
Image
300
3
VGG-16
through Conv5_3 layer
Extra Feature Layers
Classifier : Conv: 3x3x(4x(Classes+4))
Classifier : Conv: 3x3x(6x(Classes+4))
Conv: 3x3x(4x(Classes+4))
38
Conv4_3
38
512
19
Conv6
(FC6)
19
1024
19
Conv7
(FC7)
19
1024
10
Conv8_2
10
512
5
Conv9_2
5
256
3
Conv10_2
3
256
Conv11_2
256
Conv: 3x3x1024
Conv: 1x1x1024
Conv: 1x1x256
Conv: 3x3x512-s2
Conv: 1x1x128
Conv: 3x3x256-s2
Conv: 1x1x128
Conv: 3x3x256-s1
Conv: 1x1x128
Conv: 3x3x256-s1
Detections:8732 per Class
Non-Maximum Suppression
74.3mAP
59FPS
YOLO
448
Image
448
3
YOLO Customized Architecture
7
7
1024
Fully Connected
Fully Connected
7
7
30
Detections: 98 per class
Non-Maximum Suppression
63.4mAP
45FPS

Other Architectures for Object Detection

- *Mask R-CNN* (2017): Object detection + instance segmentation!
  - He, Kaiming, et al. "Mask R-CNN." Proceedings of the IEEE ICCV. 2017.
  - Independent reading: If you are interested!

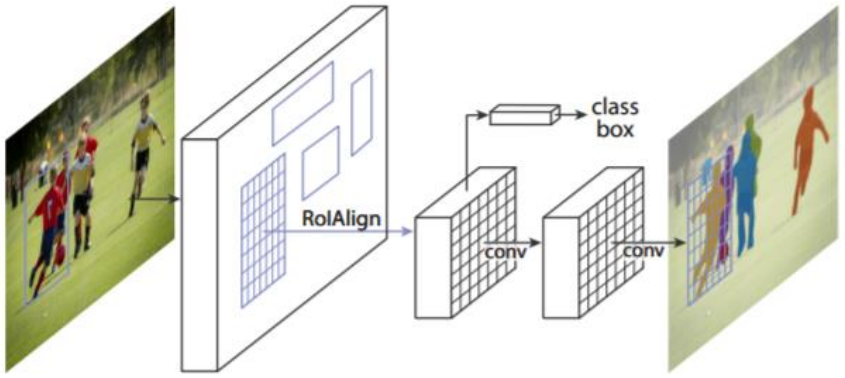


Figure 1. The **Mask R-CNN** framework for instance segmentation.

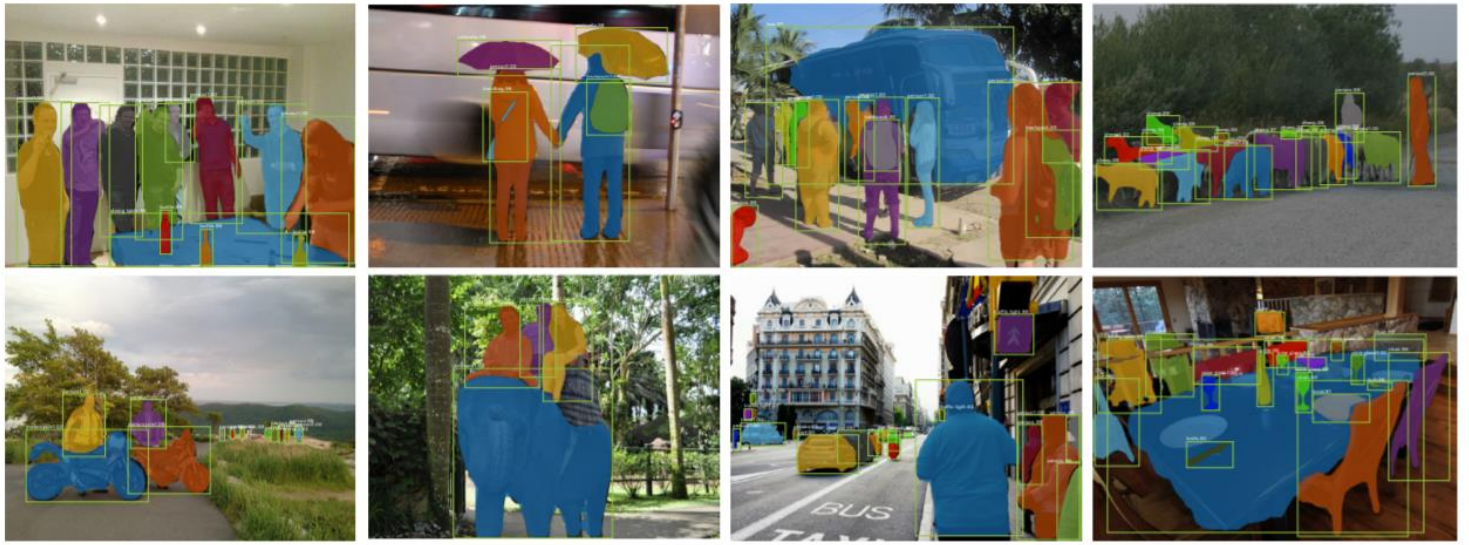


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

Image (Semantic) Segmentation

ada namanya deconvolution, dimana

dari gambar-gambar kecil itu, untuk dibesarkan lagi

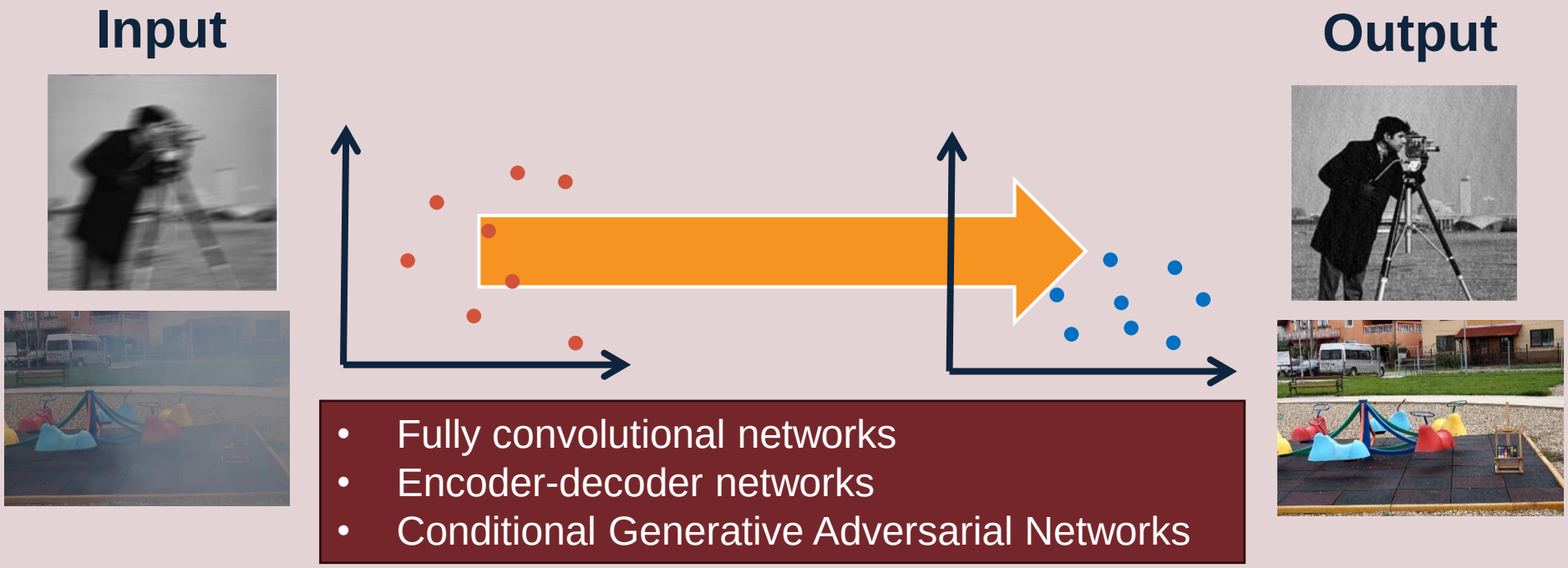
gambarnya untuk mendapatkan gambar original

- Performa image segmentation dapat dievaluasi menggunakan **precision, recall, F1-score (Dice similarity coefficient), ROC curve**, etc.

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

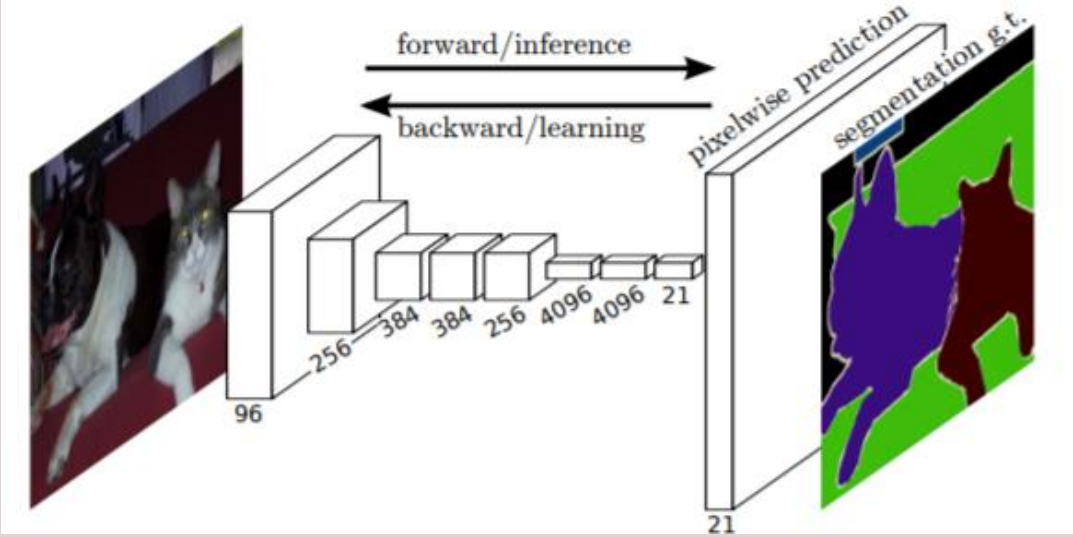
Deep Learning for Image-to-Image Translation

- By establishing the relationship between 2 image sets, we can transform/ translate images from 1 set to the other as a pixel regression problem
- Deep learning networks are very powerful to establish image to image translation – as long as if we have enough examples



Fully Convolutional Networks (FCN)

- On a CNN for a classification task, usually the final layers are fully connected
- On Fully Convolutional Networks (FCN), all layers are convolutional layers
  - Convolutions reduce spatial resolution – we need to return the spatial resolution back to the original size
  - We can use **up-sampling** or **transpose convolutional layer/ deconvolutional layers**



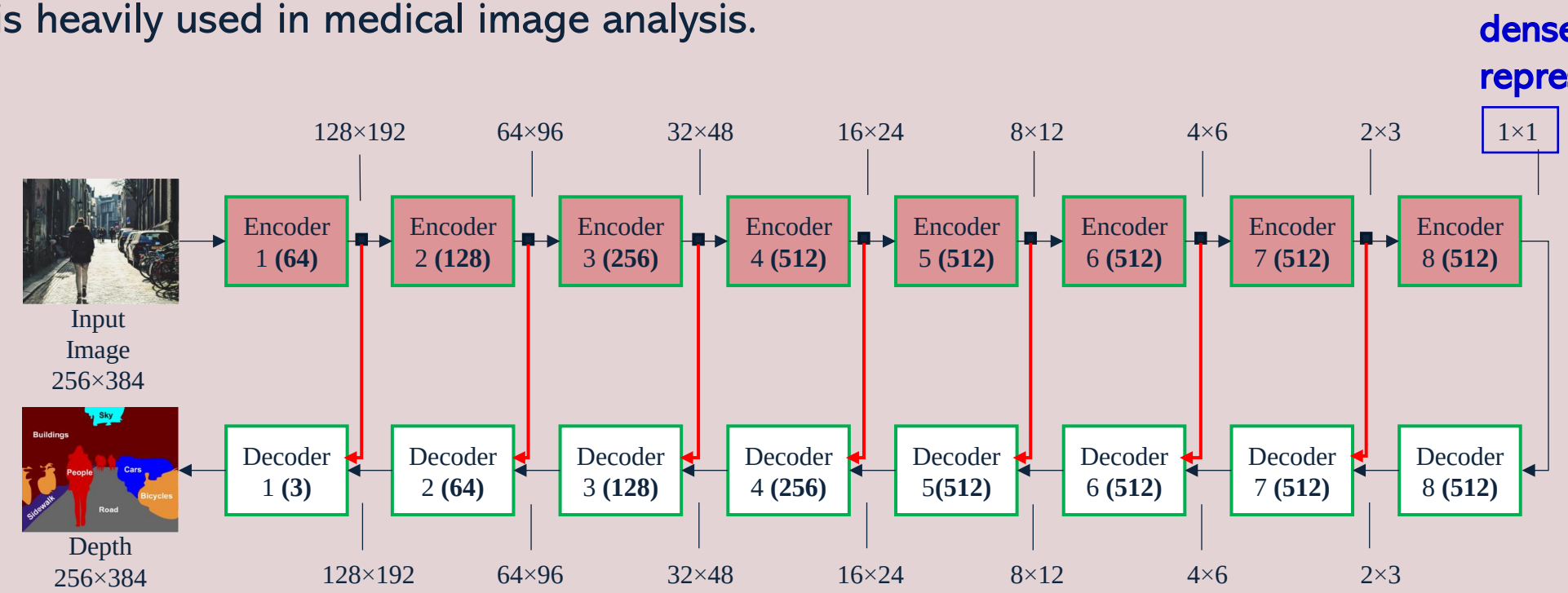
Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

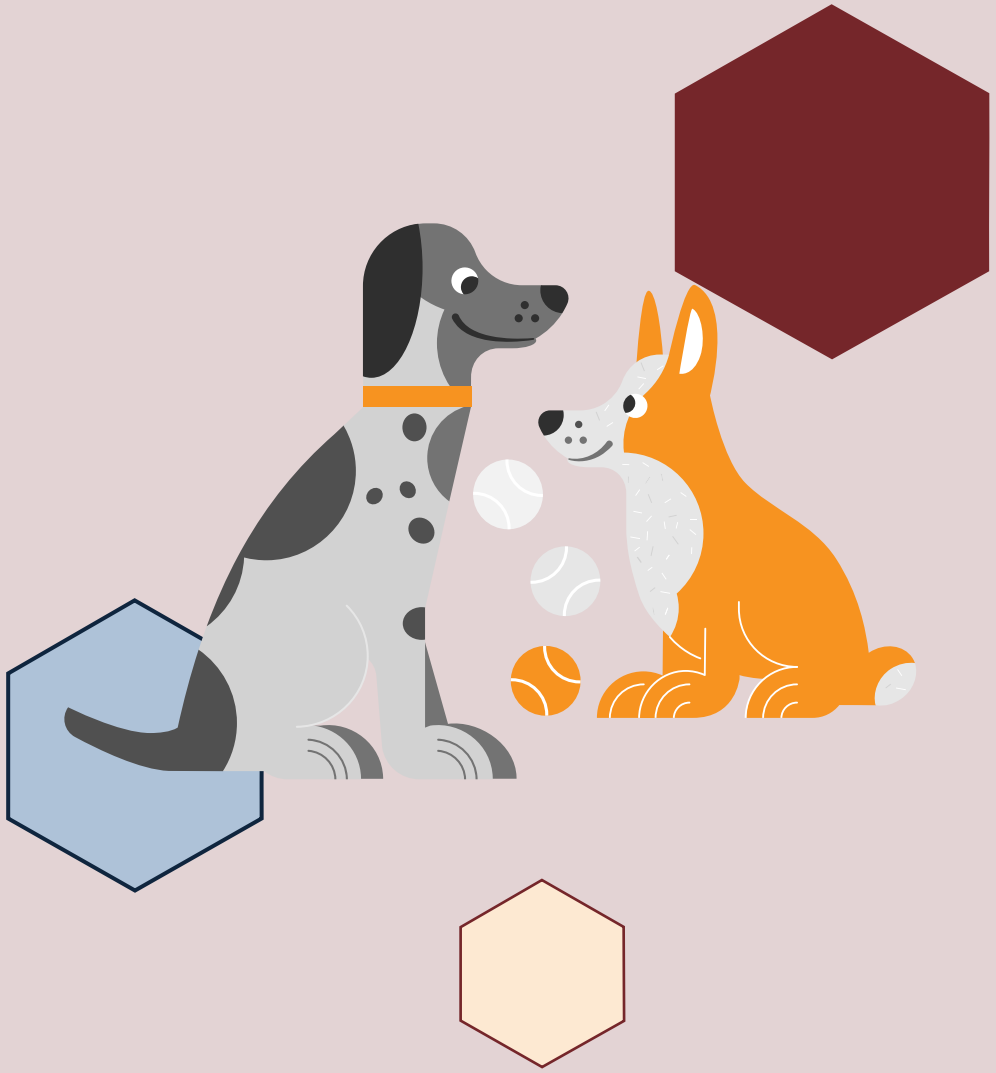
U-Net

keluar U-net, tidak hanya ada konvolusi dan dekonvolusi

Dimana disetiap layer akan menghasilkan hasil.

- U-Net expands the fully connected concepts with **encoders** (conv. + max. pooling), **dense representation**, **decoders** (conv. + up-sampling), and **skip connections**
- U-Net does domain mapping from the input image to the target (semantic segmentation).
- **Dense representation** contains the most important features of the image to map to the new domain.
- U-Net is heavily used in medical image analysis.





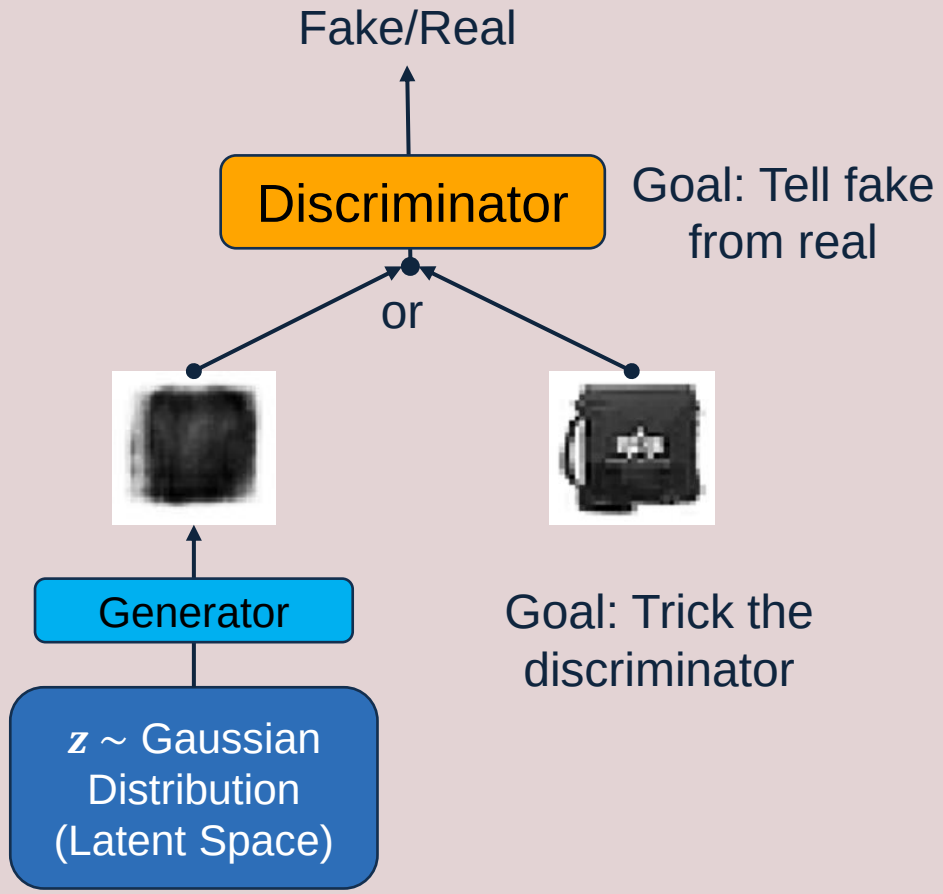
Advanced Topics of Computer Vision using Deep Learning

Section 5

Generative Adversarial Networks (GANs)

- **Generative Adversarial Networks** (GANs) consists of two types of networks that are competing against each other (adversarial)
  - **Generator** : Takes the latent representations from a distribution and generates an output image
  - **Discriminator**: Compares the generator output and an image from the training set, and distinguish which image is real (from training set) and fake (generator output).

GAN ini gunanya untuk memperoleh output yang mirip dengan yang asli.

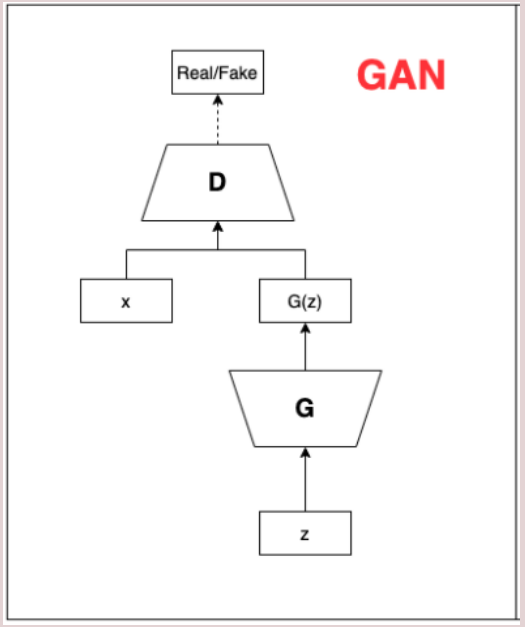


Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

Conditional GAN

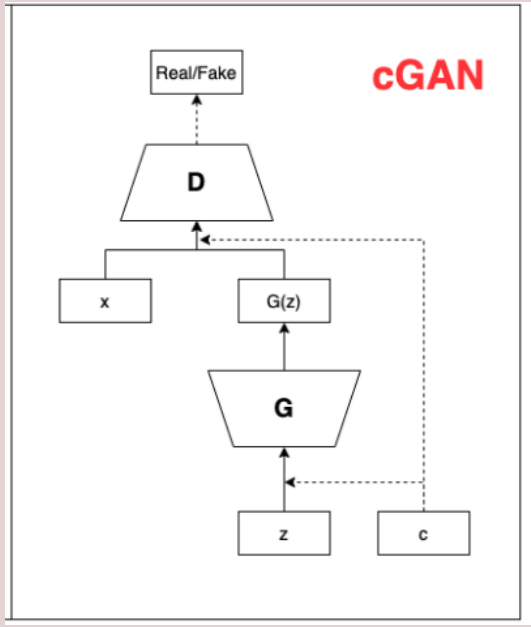
- Generative Adversarial Networks

"….are based on a game scenario in which the generator network competes against an adversary. The generator network directly produces samples. Its adversary, the discriminator network, attempts to distinguish between samples drawn from the training data and samples drawn from the generator."[1]



- Conditional GAN

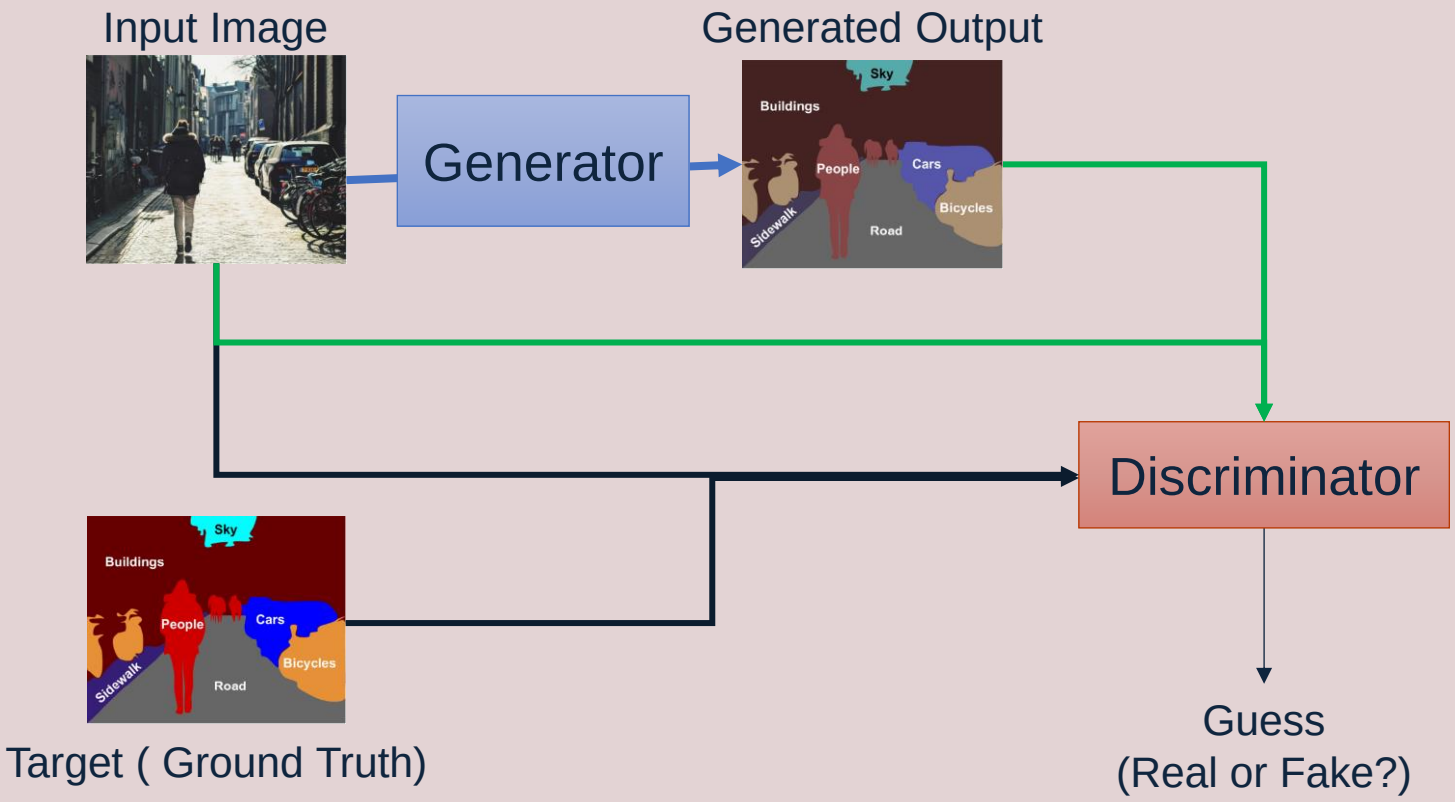
"..if both the generator and discriminator are conditioned on some extra information y. y could be any kind of information, such as class labels or data from other modalities. We can perform the conditioning by feeding y into the both the discriminator and generator as [an] additional input layer"[2]

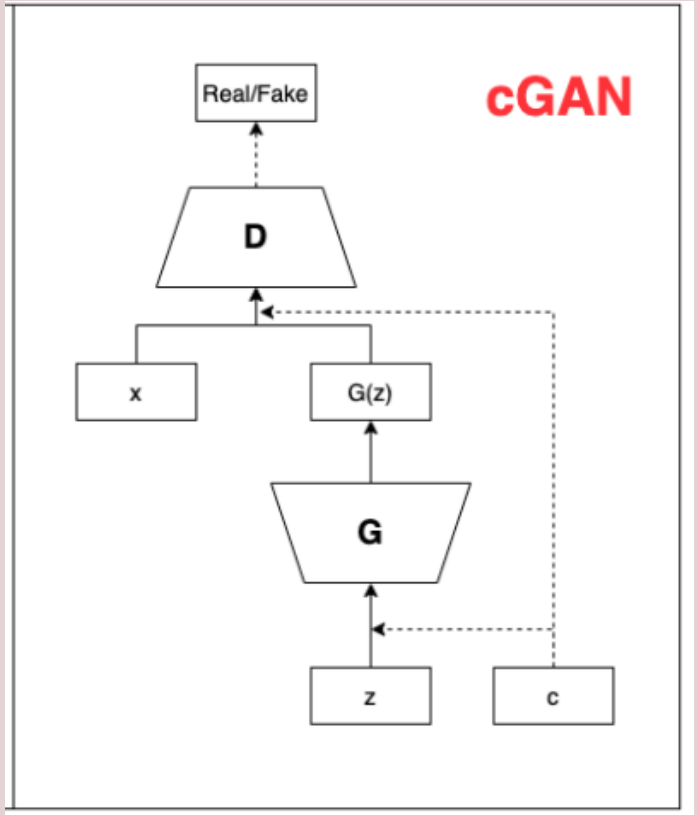


[1]Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
[2]Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).

Conditional GAN for Image-to-Image Translation

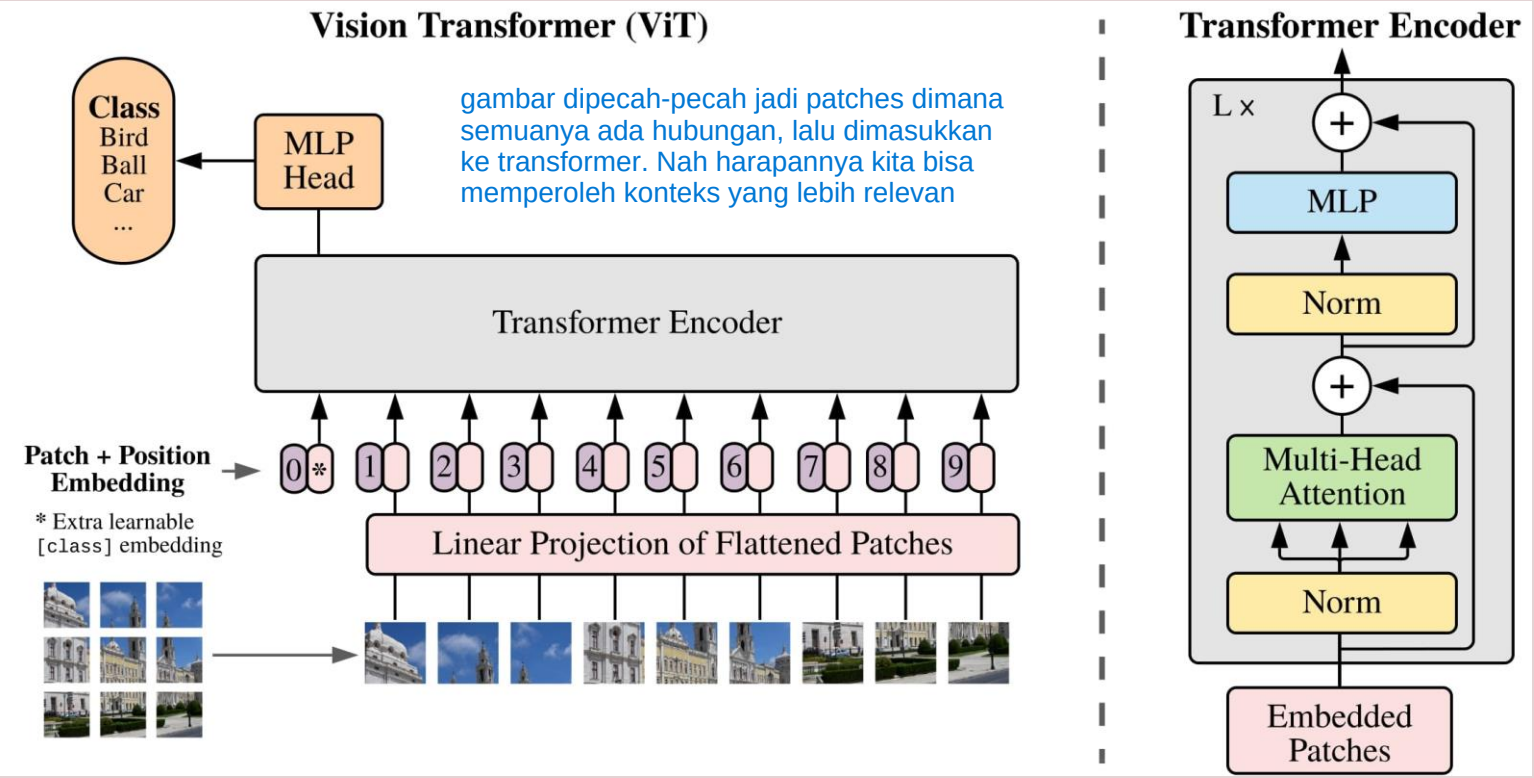
- The extra information y to guide the generator can be the input image
- The generator tries to fool the discriminator, the discriminator tries to distinguish the real and fake pairs (adversaries)





Vision Transformer (ViT)

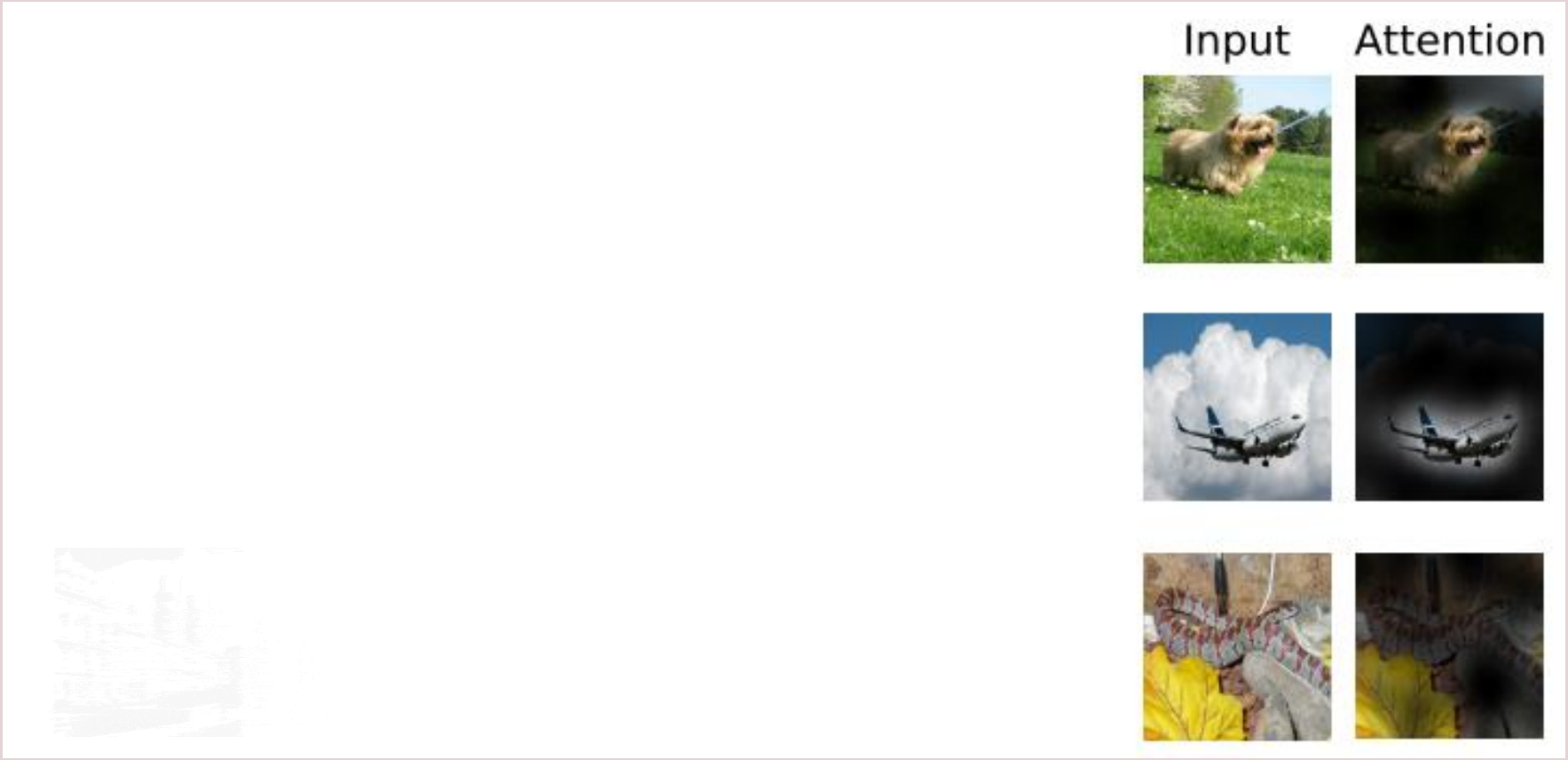
- Attention mechanisms combined with RNNs were the predominant architecture for facing any **task involving text** until 2017, when a paper was published and changed everything, giving birth to Transformers.
- Expanded into image domain: Takes image patches, not words



Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

Vision Transformer (ViT) in Action

- Vision transformers have extensive applications in popular image recognition tasks such as object detection, image segmentation, image classification, and action recognition.



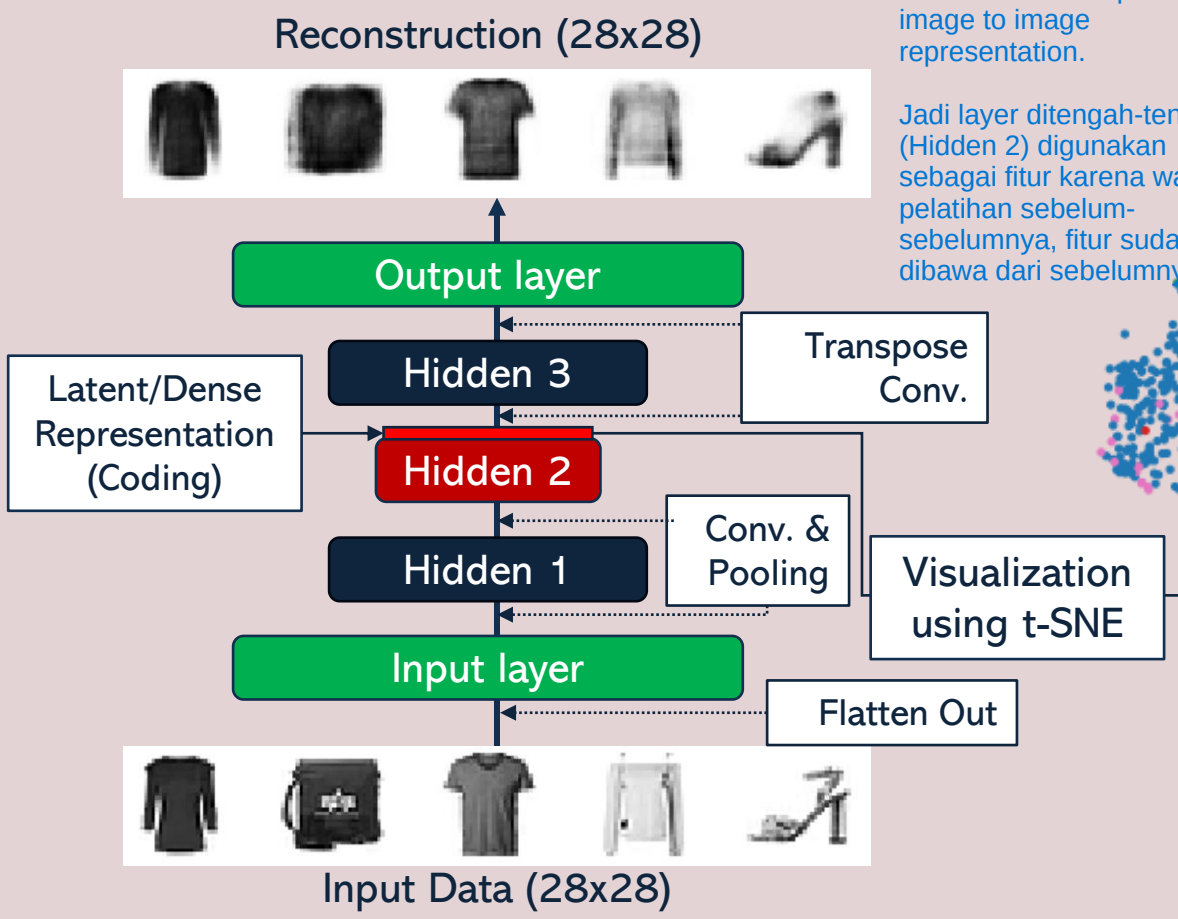
Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

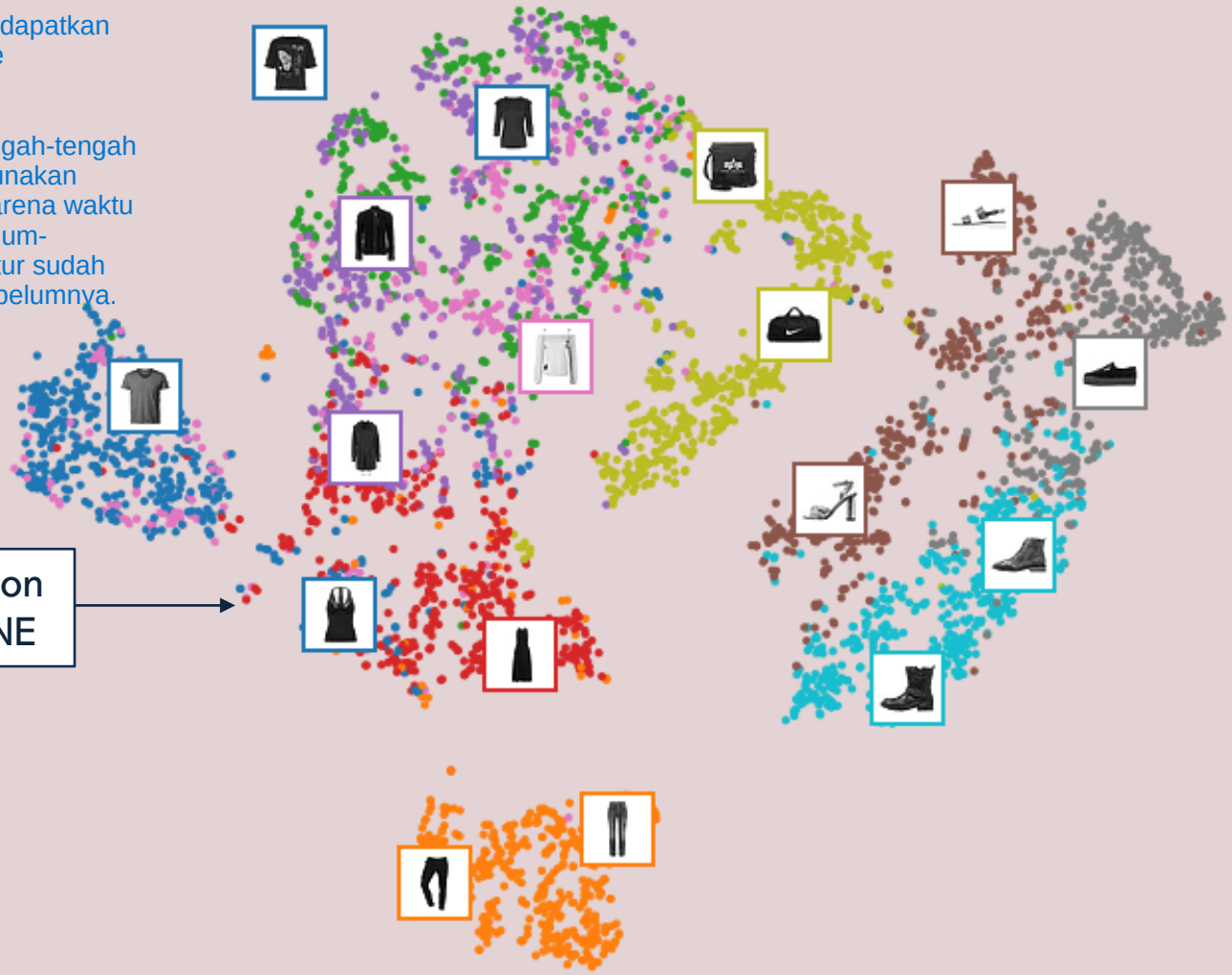
Latent/Dense Representation of Images

How to use t-SNE effectively: https://distill.pub/2016/misread-tsne/

waktu kita mendapatkan image to image representation.

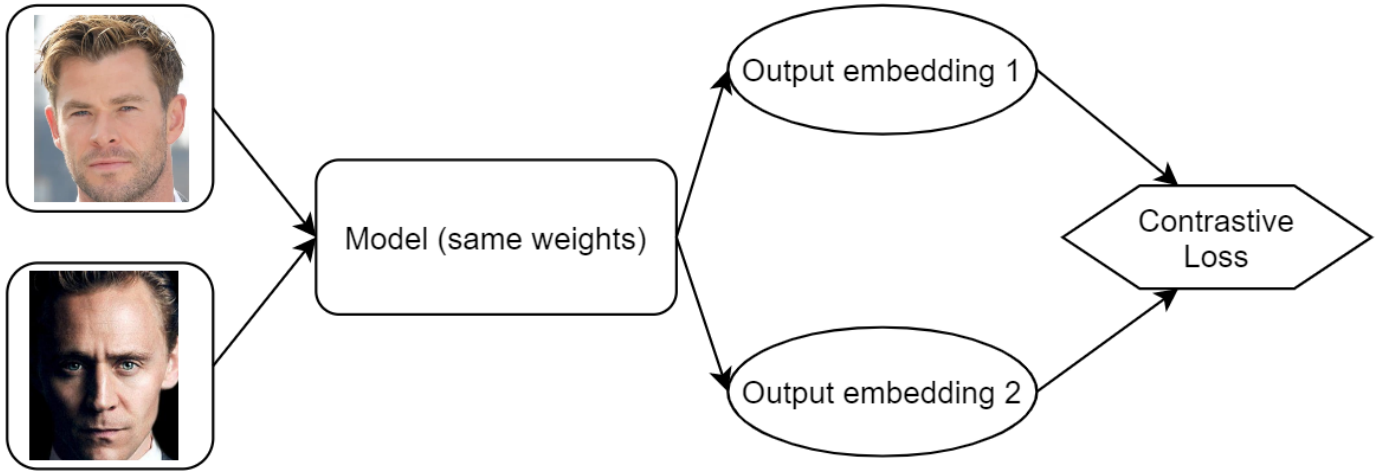
Jadi layer ditengah-tengah (Hidden 2) digunakan sebagai fitur karena waktu pelatihan sebelum-sebelumnya, fitur sudah dibawa dari sebelumnya.





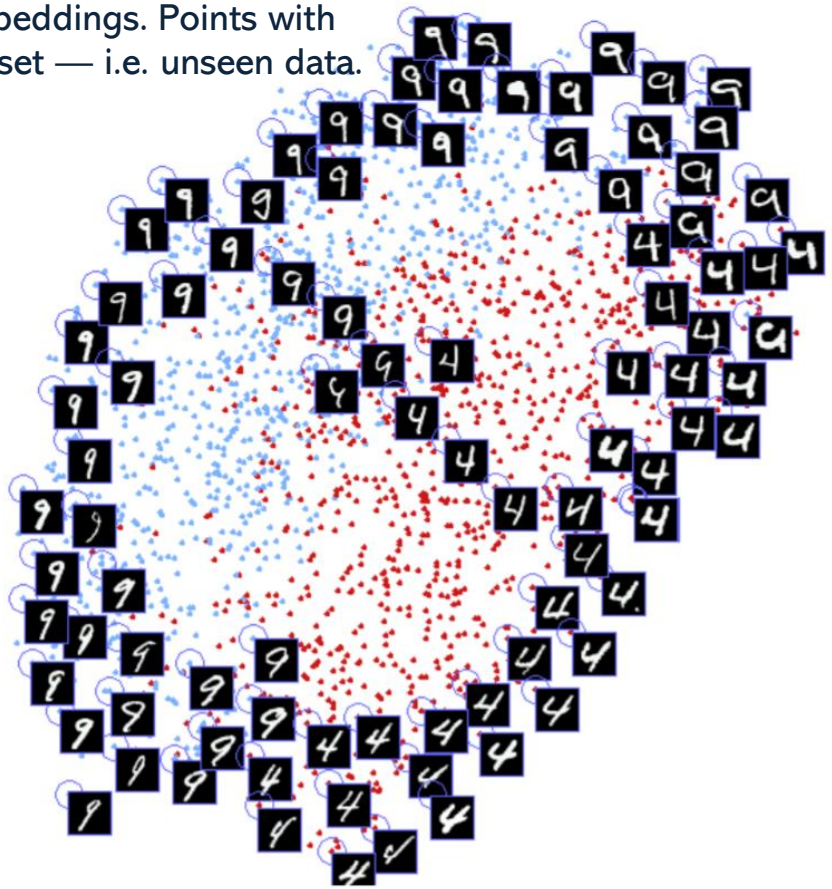
Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms." *arXiv preprint arXiv:1708.07747* (2017).

Distance between Two (or Multiple) Inputs in the Latent Space



Siamese network architecture

Distribution of MNIST images embeddings. Points with images attached are from test dataset — i.e. unseen data.

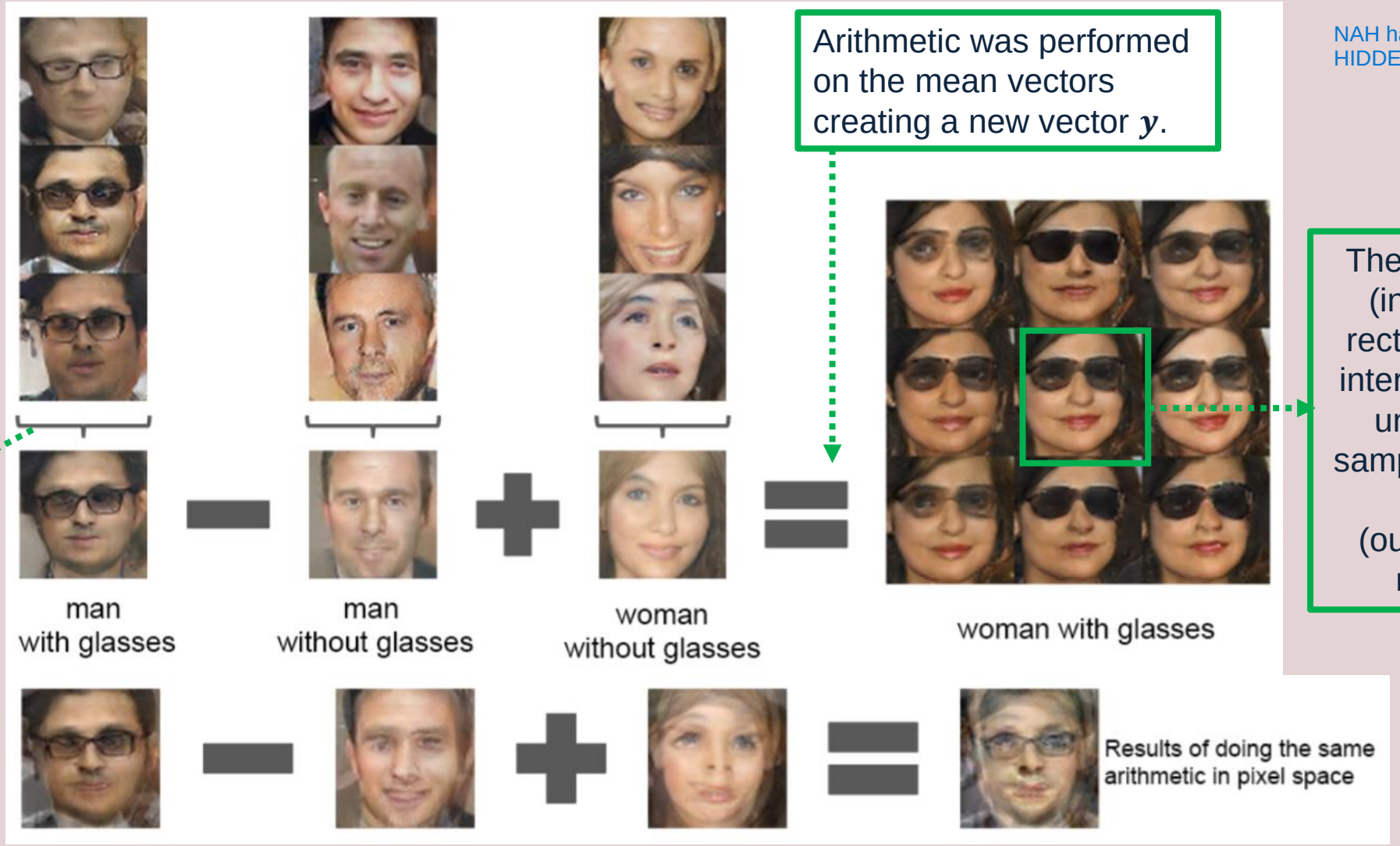


Further reading: Understanding Ranking Loss, Contrastive Loss, Margin Loss, Triplet Loss, Hinge Loss and all those confusing names. (https://gombru.github.io/2019/04/03/ranking_loss/)

Figures are from Losses explained: Contrastive Loss (https://medium.com/@maksym.bekuzarov/losses-explained-contrastive-loss-f8f57fe32246)

Arithmetic in Latent Space using GANs

NAH hal ini dilakukan semuanya di HIDDEN LAYER 2.




Thank you!

CSCE604133 Computer Vision
Faculty of Computer Science
Universitas Indonesia

Dr. Eng. Laksmita Rahadianti
Muhammad Febrian Rachmadi, Ph.D.
Dr. Dina Chahyati, Prof. Dr. Aniati M. Arymurthy