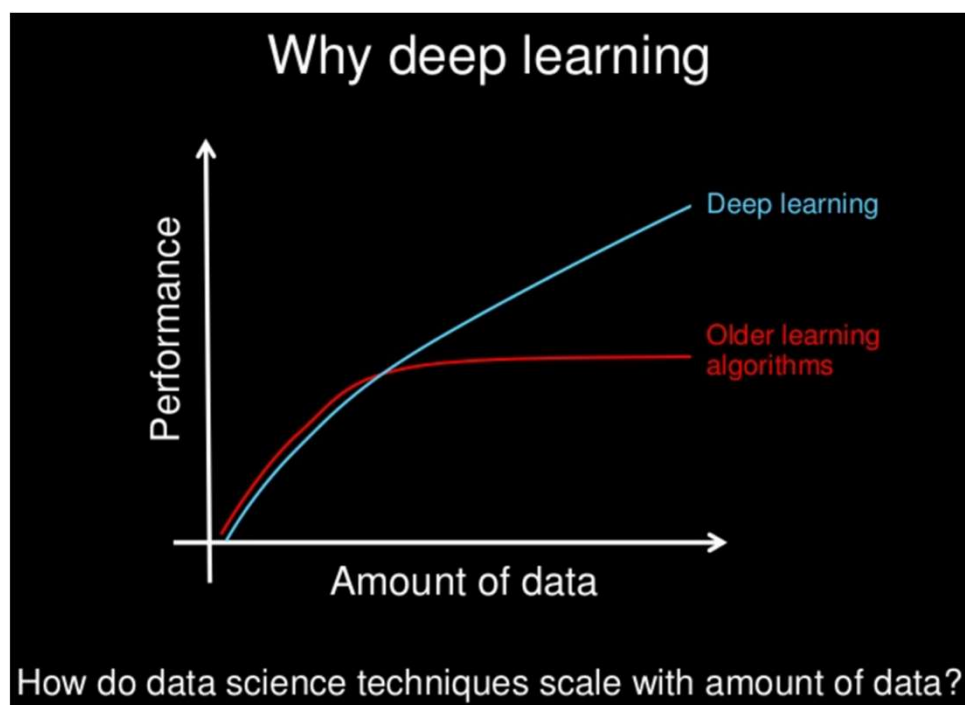


Day 4 – Deep Learning Keras

Agenda

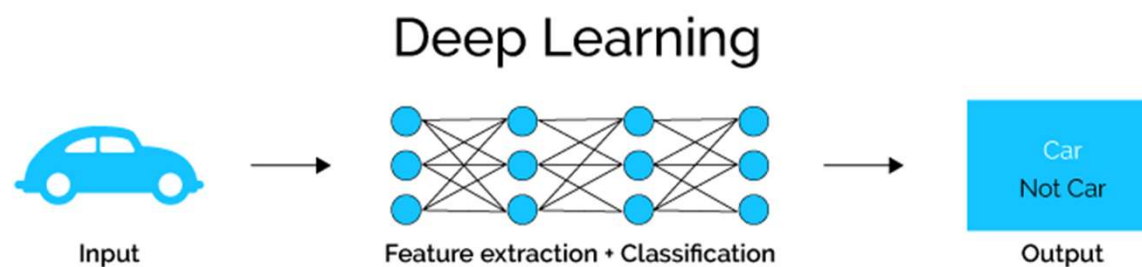
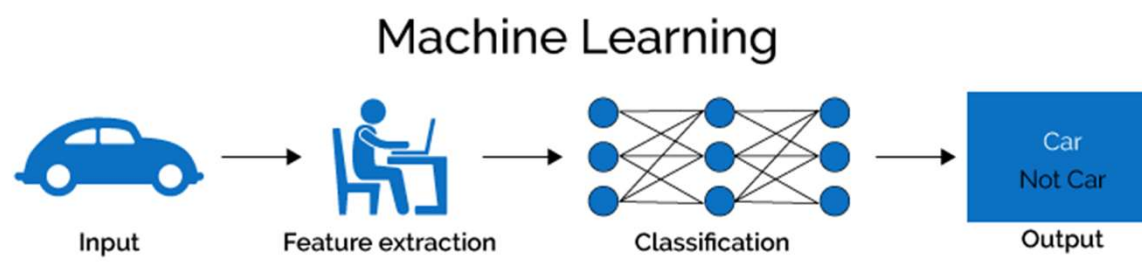
- ช่วงเช้า
 - Intro to deep learning with Keras
 - Lab
 - การใช้ deep learning กับข้อมูลประเภท tabular data
- ช่วงบ่าย
 - การ deploy model เป็น REST API (own server)
 - การ deploy บน cloud ml engine
 - automl

Deep Learning (DL)



<https://machinelearningmastery.com/what-is-deep-learning/>

ML vs. DL



<https://semiengineering.com/deep-learning-spreads/>

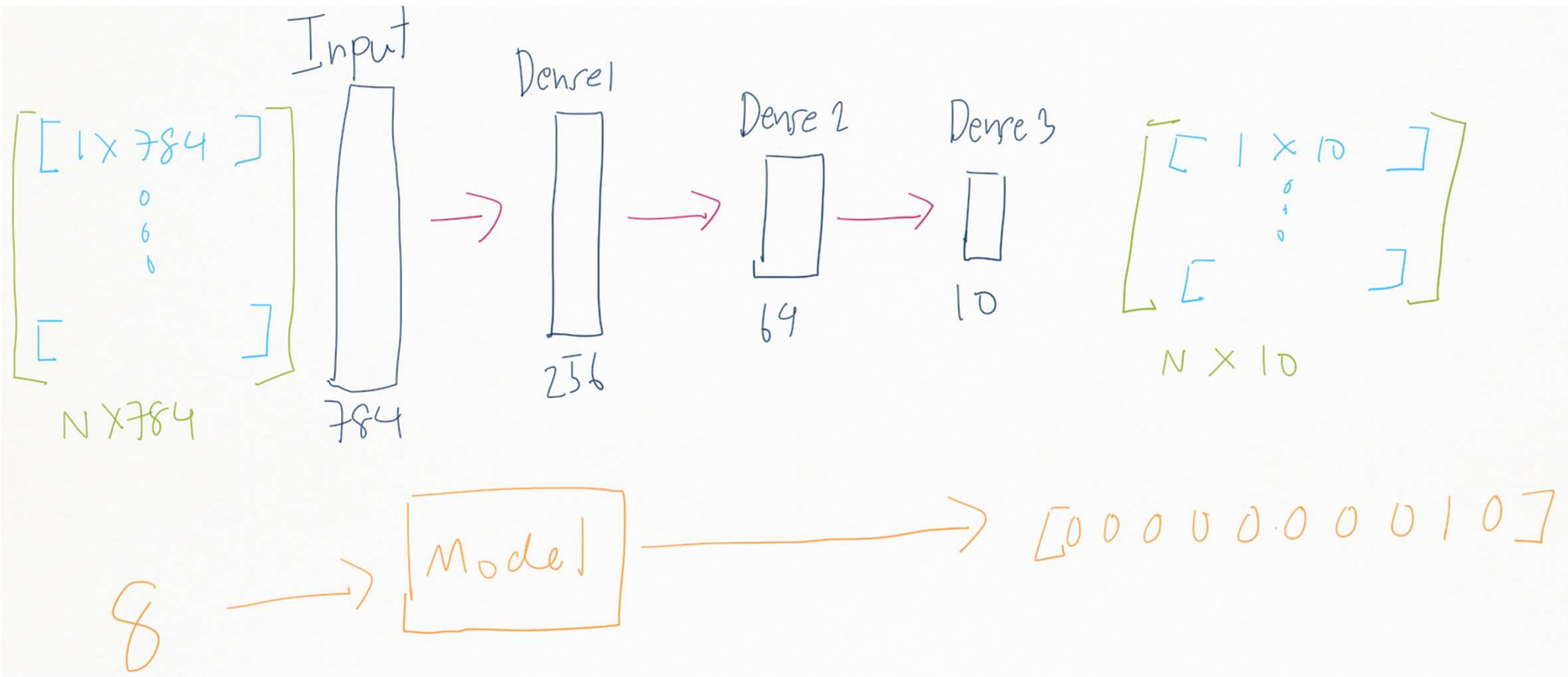
Keras vs. Tensorflow

- tf เป็น low level
- Keras เป็น high-level API ครอบ
- Keras สามารถใช้กับ framework Theano, CNTK ได้ด้วย
- tf version ใหม่ๆรวม Keras เข้ามาด้วยเป็น tf.keras และใช้ tf เป็น backend
- การใช้งานทั่วไป Keras สะดวกกว่าเขียน low-level tf มาก

Example 01 – MNIST

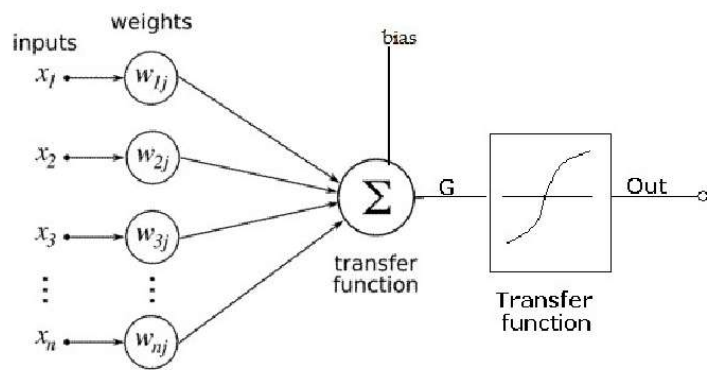
- 28 x 28 gray scale
- 10 class: 0 to 9



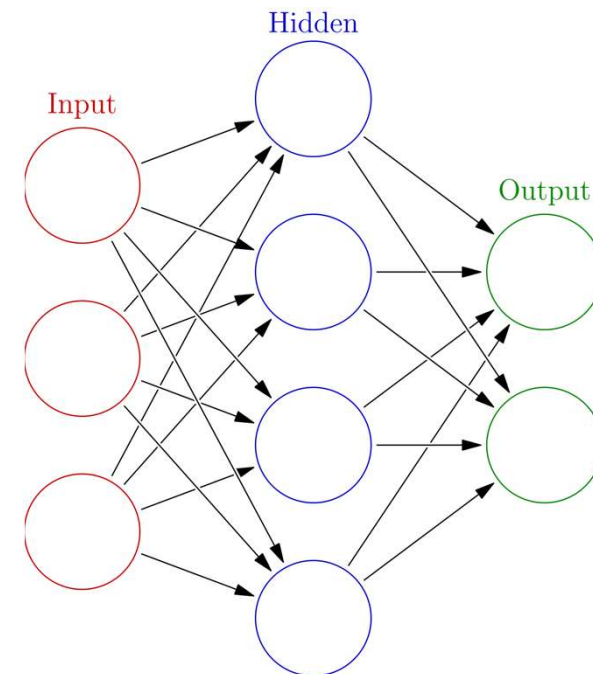


Dense layer

- ทุก unit ในเลเยอร์ N ต่อกับทุก unit ในเลเยอร์ก่อนหน้า



https://www.researchgate.net/figure/Artificial-Neuron-model_fig4_277774116

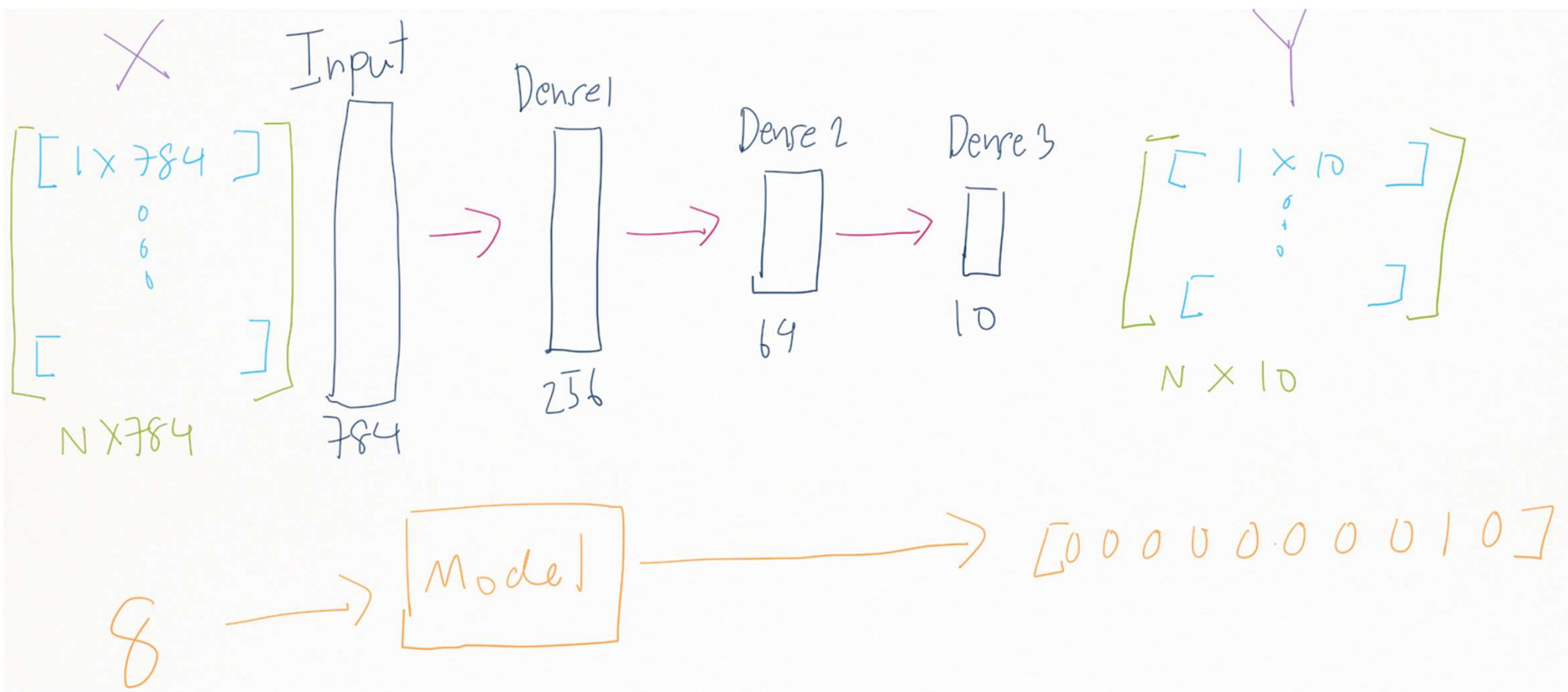


https://en.wikipedia.org/wiki/Artificial_neural_network

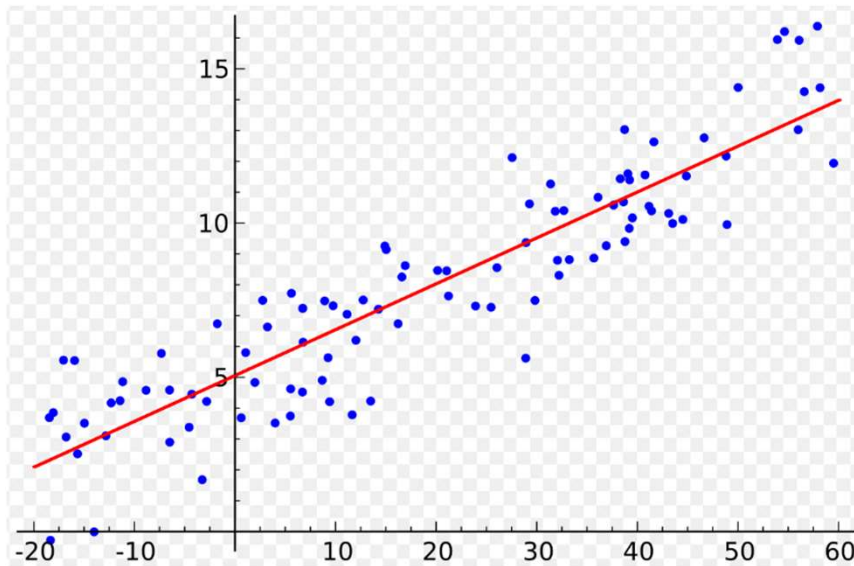
Neural network (NN)

$$\min_{\theta} = J(f_{\theta}(X), Y)$$

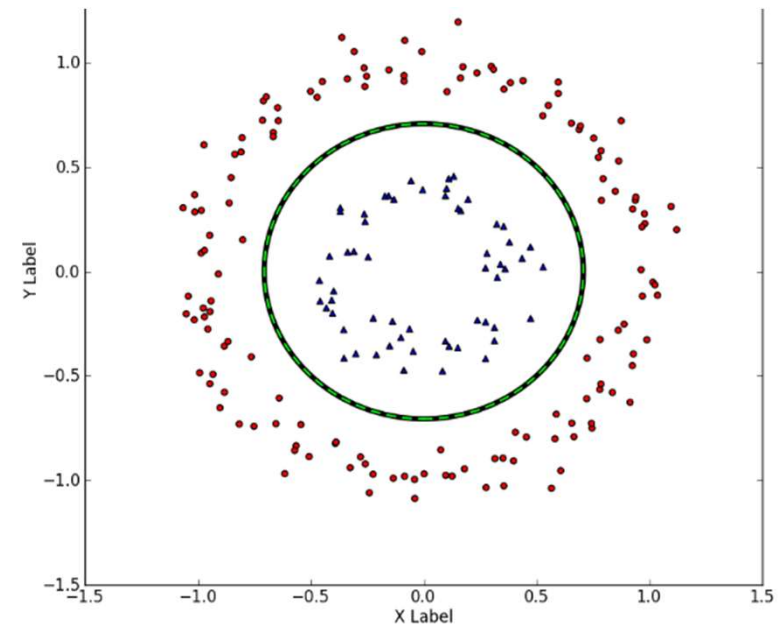
- X inputs
- Y outputs
- f_{θ} function จาก input \rightarrow output
- J “loss” function
- ปรับ θ เพื่อให้ $f_{\theta}(X)$ เข้าใกล้ Y . โดยการทำให้ J ลดลงให้น้อยที่สุด (minimize)
- θ คือ weight ทั้งหมดของ NN.



Regression vs. classify



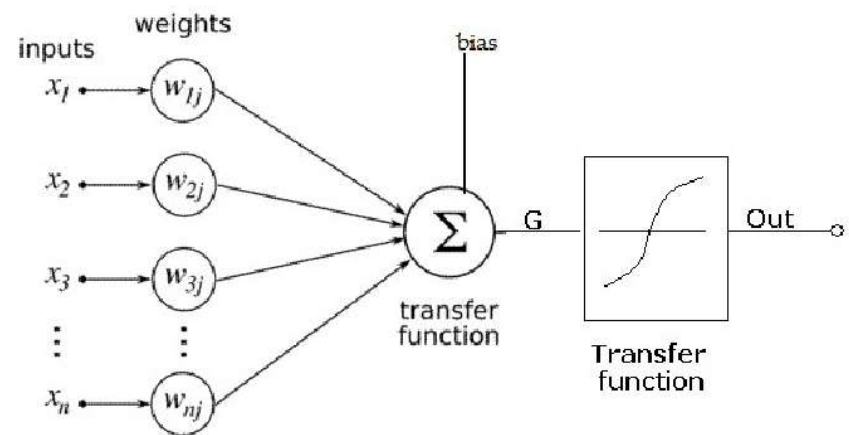
https://en.wikipedia.org/wiki/Linear_regression



https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Activation function

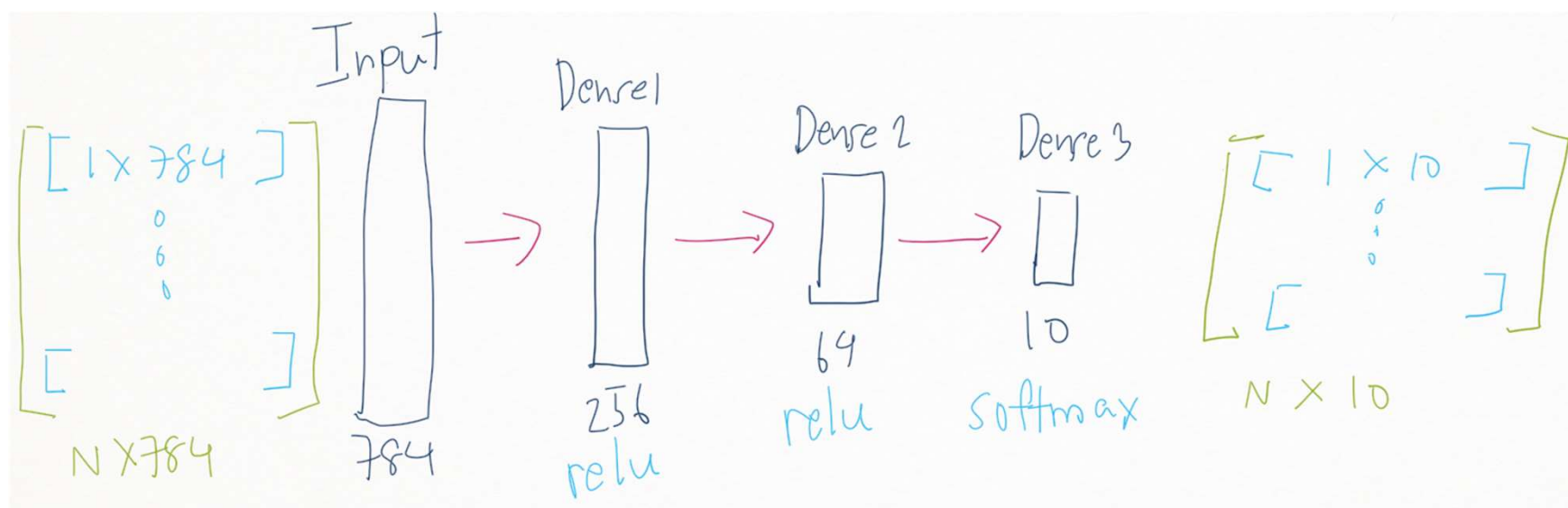
- linear
- tanh
- relu
- sigmoid
- softmax



https://www.researchgate.net/figure/Artificial-Neuron-model_fig4_277774116

Categorical vs. continuous

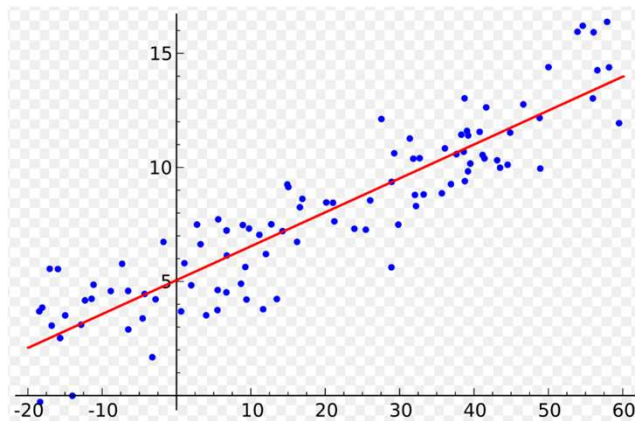
NN for Classify



Loss function J

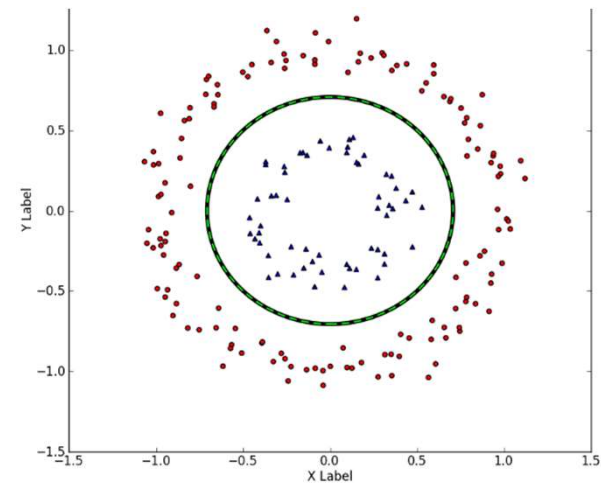
Regression:

- $MSE(\hat{Y}, Y)$



Classify

- $cross_entropy(\hat{Y}, Y)$



01 - IntroKeras.ipynb

Example 02 – Boston Housing

train - Excel															
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do															
A1															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	ID	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
2	1	0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
3	2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
4	4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
6	7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
7	11	0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15
8	12	0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9
9	13	0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7
10	14	0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4
11	15	0.63796	0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21	380.02	10.26	18.2
12	16	0.62739	0	8.14	0	0.538	5.834	56.5	4.4986	4	307	21	395.62	8.47	19.9
13	17	1.05393	0	8.14	0	0.538	5.935	29.3	4.4986	4	307	21	386.85	6.58	23.1
14	19	0.80271	0	8.14	0	0.538	5.456	36.6	3.7965	4	307	21	288.99	11.69	20.2
15	21	1.25179	0	8.14	0	0.538	5.57	98.1	3.7979	4	307	21	376.57	21.02	13.6
16	22	0.85204	0	8.14	0	0.538	5.965	89.2	4.0123	4	307	21	392.53	13.83	19.6
17	23	1.23247	0	8.14	0	0.538	6.142	91.7	3.9769	4	307	21	396.9	18.72	15.2
18	24	0.98843	0	8.14	0	0.538	5.813	100	4.0952	4	307	21	394.54	19.88	14.5
19	28	0.95577	0	8.14	0	0.538	6.047	88.8	4.4534	4	307	21	306.38	17.28	14.8
20	31	1.13081	0	8.14	0	0.538	5.713	94.1	4.233	4	307	21	360.17	22.6	12.7

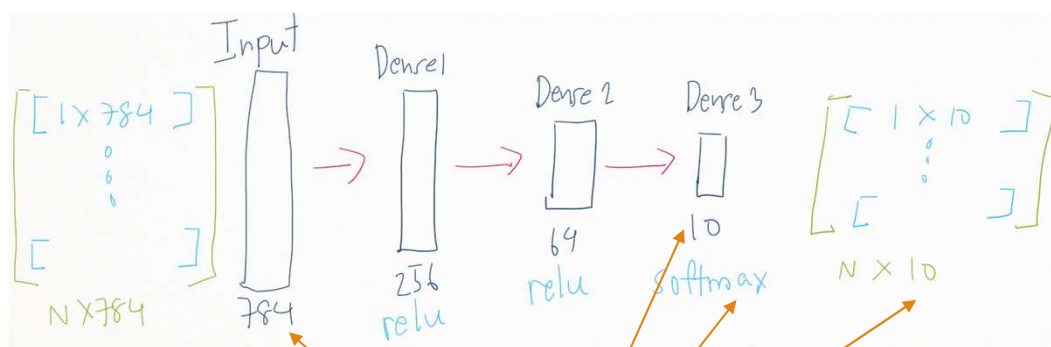
Model for example 02

- สิ่งี่เปลี่ยนจากตัวอย่างที่ 1
 - ขนาดของ output: 10 -> 1
 - Activation function ของ output layer (dense3): softmax -> linear
 - Loss: categorical_crossentropy -> mean_squared_error
 - Matric: accuracy -> mse
- ข้อสังเกต
 - ควรทำการ normalize feature ก่อนทำการเทรน

การทำ Normalize feature

$$f = \frac{f - \mu_f}{sd(f)}$$

Model for example 02



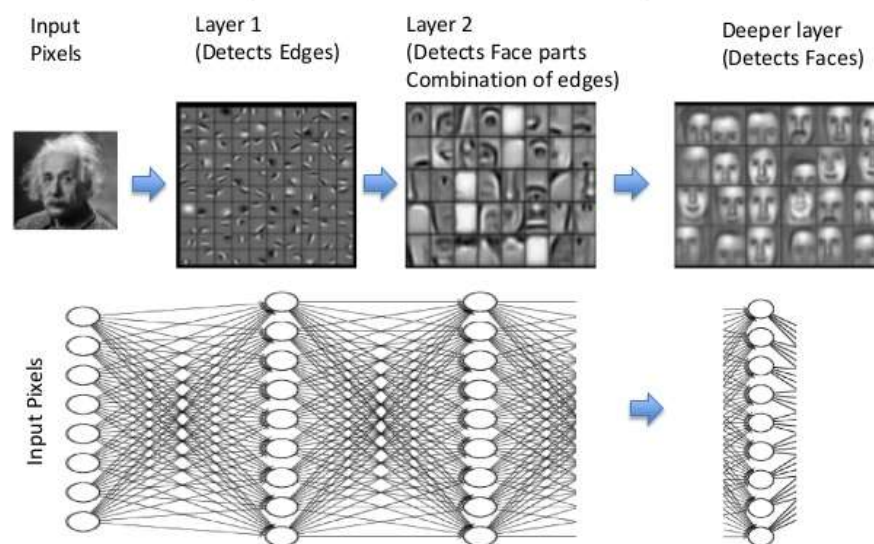
จุดที่เปลี่ยน

Lab 01 – breast cancer data

<https://archive.ics.uci.edu/ml/datasets/breast+cancer>

ทำไมต้องมีหลายเลเยอร์?

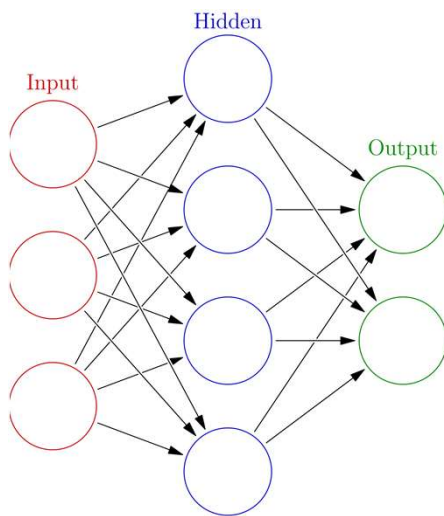
Feature Learning/Representation Learning (Ex. Face Detection)



<https://www.slideshare.net/RukshanBatuwita/deep-learning-towards-general-artificial-intelligence>

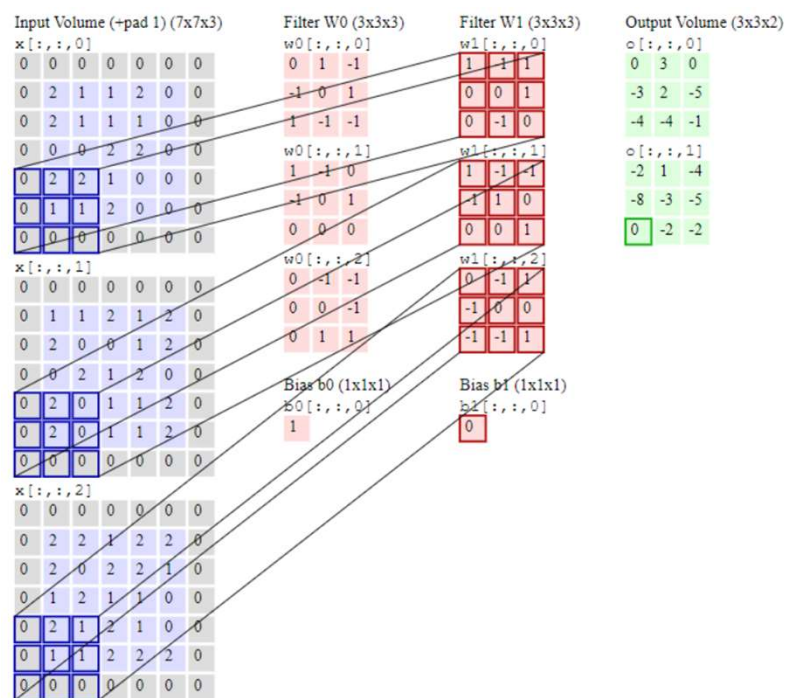
Convolutional Neural Network (CNN)

Layer แบบ dense นั้นถ้ามีจำนวนเลเยอร์ที่มาก จำนวน weight จะมากเกินไป



- Input = 1M pixel image
- Dense เลเยอร์ที่มีจำนวน 100 ยูนิต = 100M weights
- การต่อทุก pixel เข้ากับทุก unit ไม่เหมาะสมกับข้อมูลแบบภาพ ที่มีข้อมูล local เฉพาะบริเวณใดๆ

Convolutional Layer



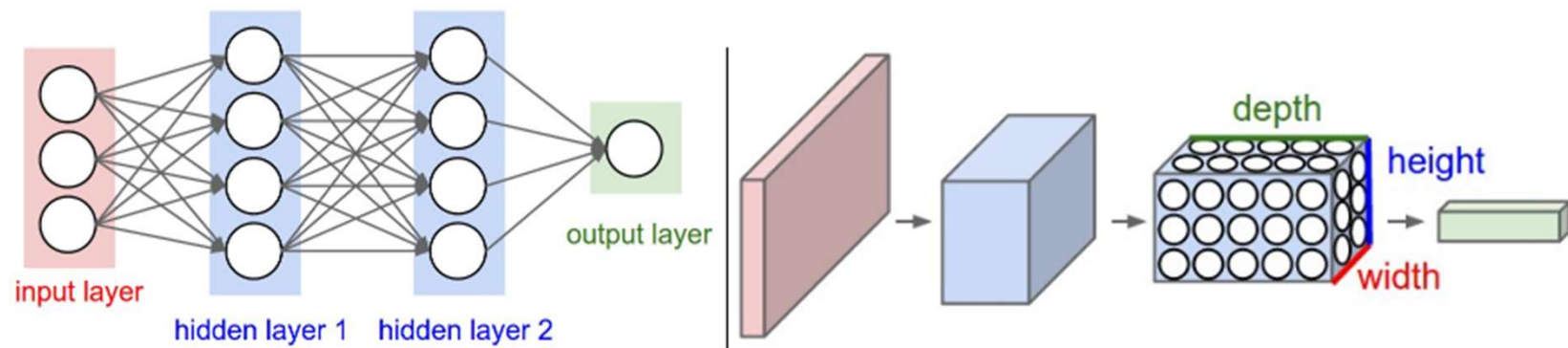
- จำนวน channel ของ filter เท่ากับ channel ของ input
- จำนวน channel ของ output ทำกับจำนวน filter
- ขนาดกว้าง-ยาวของ output ขึ้นอยู่กับ stride และ padding

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

Convolutional Layer

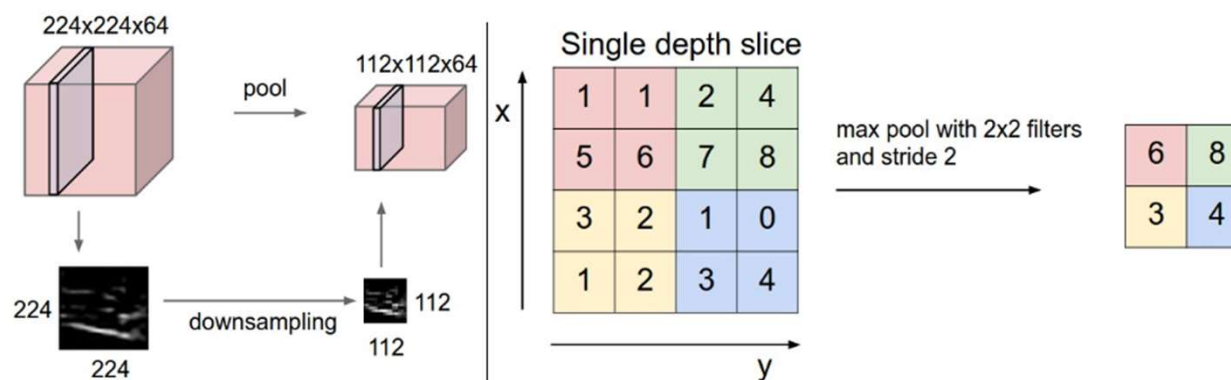
- spec ของ 1 convolutional layer (conv) ประกอบด้วย
 - จำนวน filter/kernel
 - ขนาดของ kernel
 - ขนาด stride
 - วิธีการ padding

Conv vs. dense



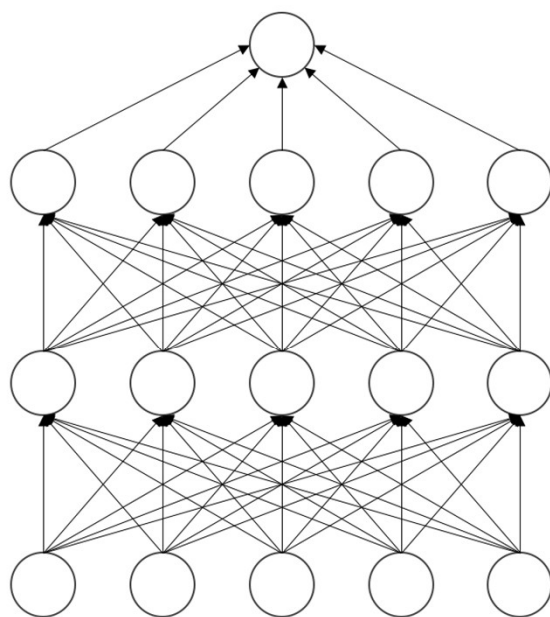
<http://cs231n.github.io/convolutional-networks/>

Max pooling

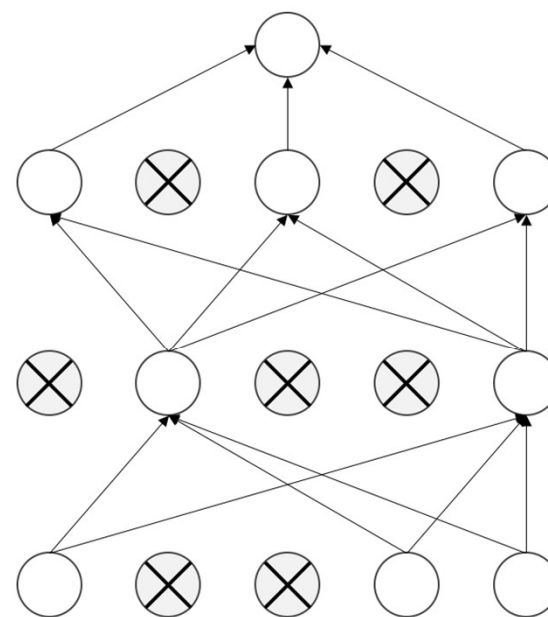


<http://cs231n.github.io/convolutional-networks/>

Dropout



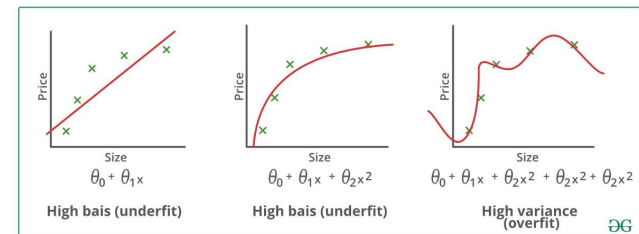
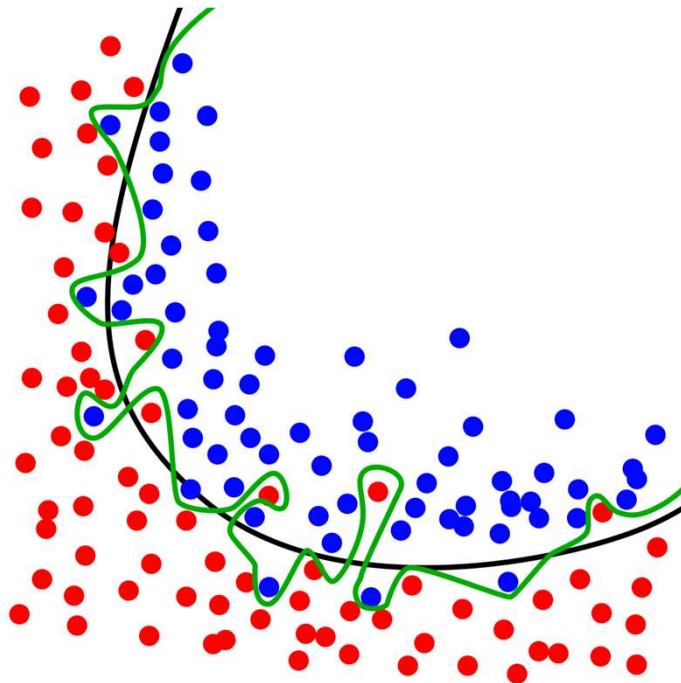
Standard Neural Net



After applying dropout

<https://www.oreilly.com/library/view/deep-learning-for/9781788295628/a22e6b18-79e3-4875-b003-2f4c6080bf54.xhtml>

Overfitting



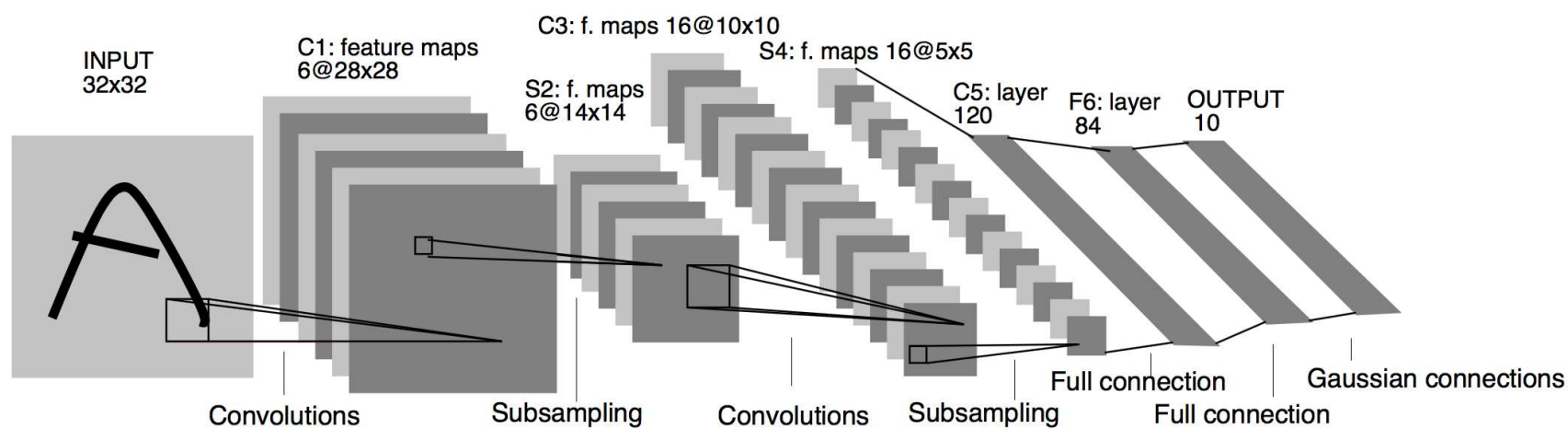
<https://en.wikipedia.org/wiki/Overfitting>

<https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>

Overfitting

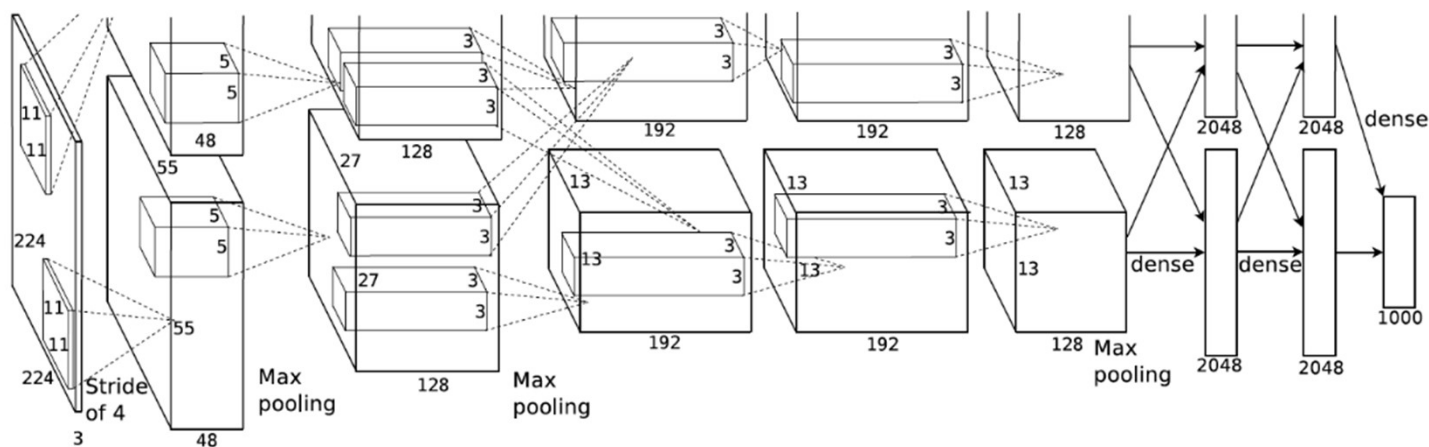


Lenet



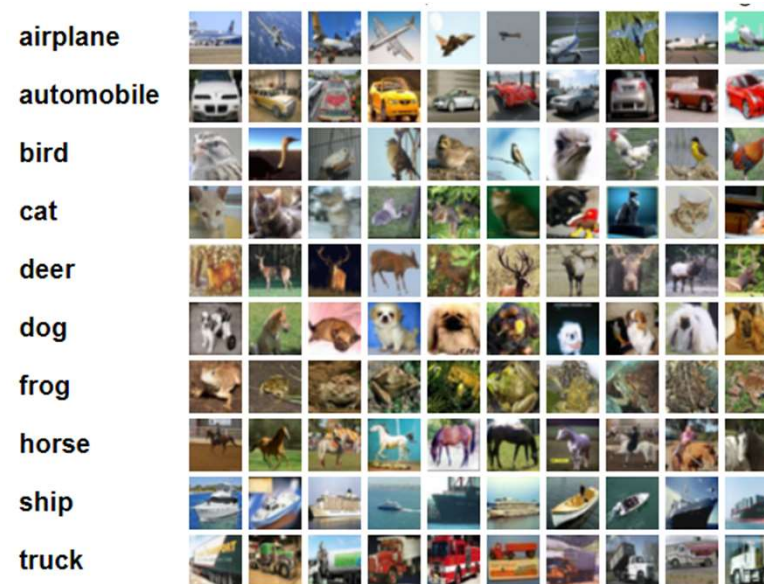
LeCun, Yann, et al. "Gradient-based learning applied to document recognition."
Proceedings of the IEEE 86.11 (1998): 2278-2324.

Alexnet



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton.
 "Imagenet classification with deep convolutional neural networks."
Advances in neural information processing systems. 2012.

Example 03 – cifar10 with CNN



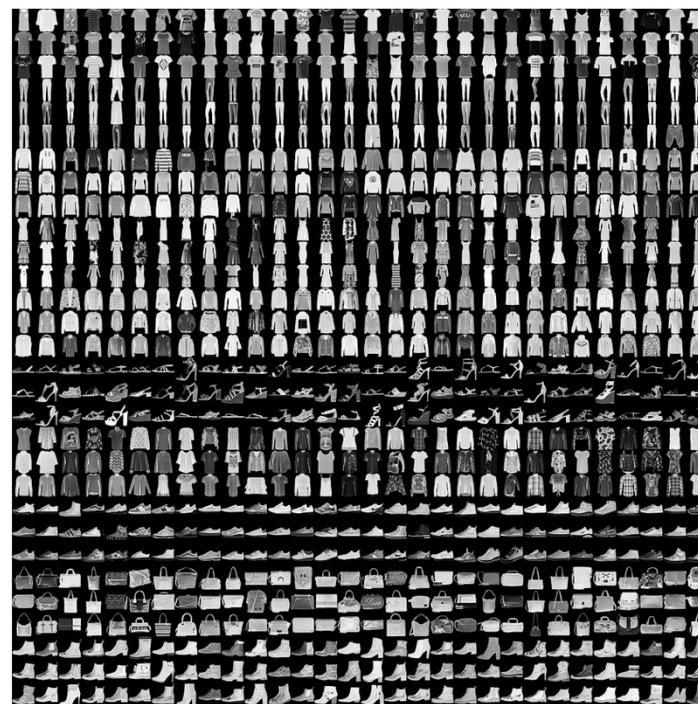
<https://www.cs.toronto.edu/~kriz/cifar.html>

Lab 02 – fashion mnist CNN

Fashion mnist

```
from keras.datasets import  
fashion_mnist
```

```
(trainX, trainY), (testX, testY) =  
fashion_mnist.load_data()
```



<https://github.com/zalando-research/fashion-mnist>

Image recognition ด้วยดาต้าของตนเอง

- จากตัวอย่าง mnist, cifar10, fashion mnist เป็น built-in dataset ของ Keras ทั้งหมด
- หากต้องการใช้ดาต้าที่เป็นภาพเก็บไว้ในไฟล์ของเราเอง จะทำอย่างไร?
- ใช้ ImageDataGenerator

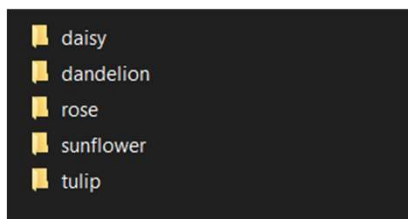
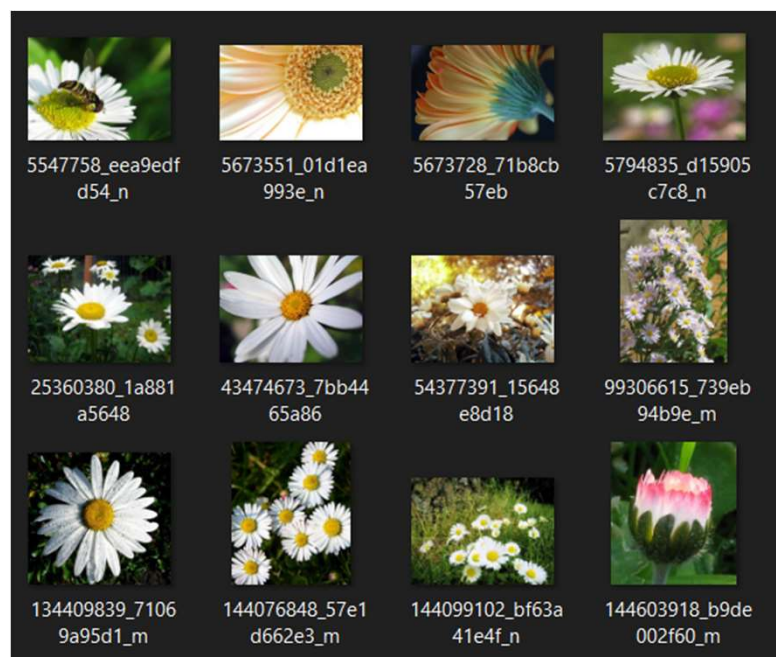
Train /

- Class1 / <- รูปของ class 1
- Class 2 /
- ...
- Class n /

Test /

- Class1 /
- Class 2 /
- ...
- Class n /

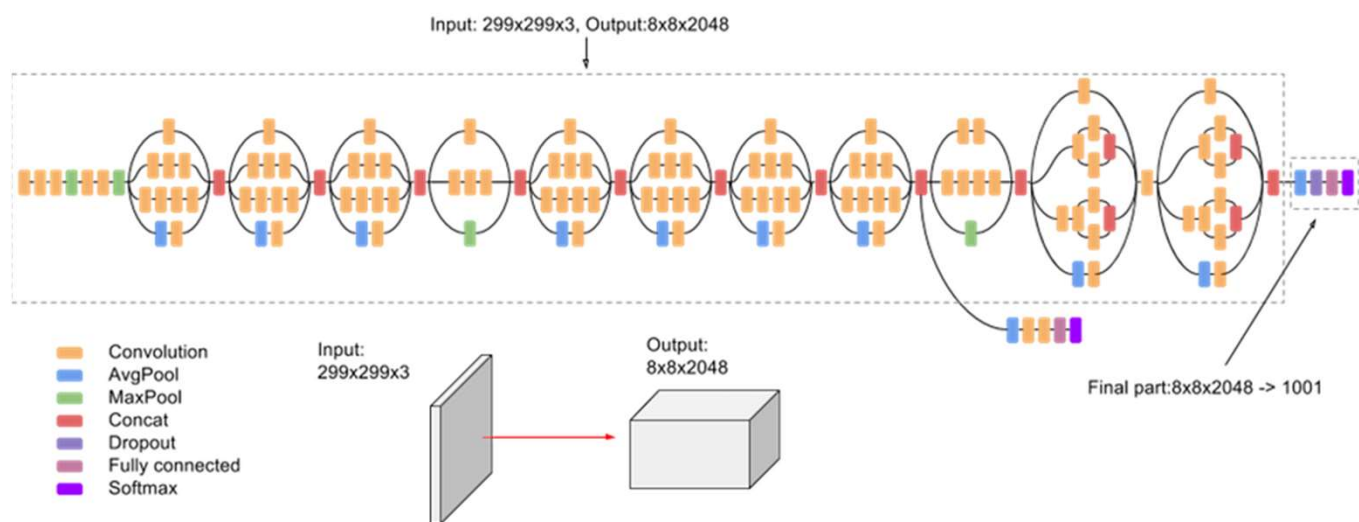
Example 04 – flower data



Lab 03 – blood cell

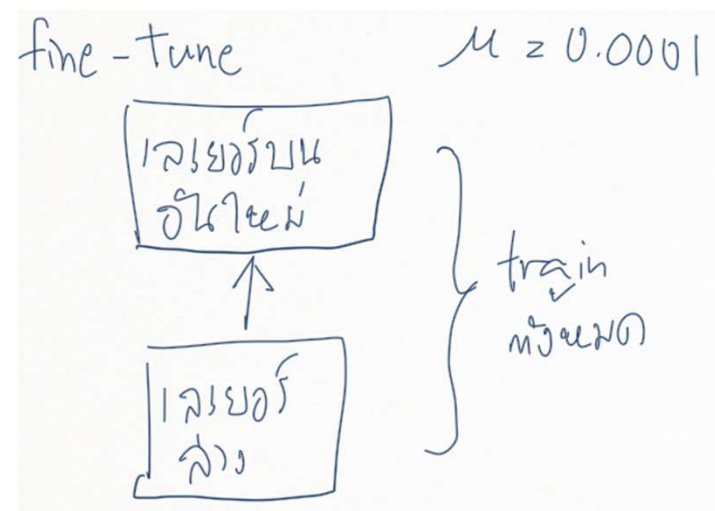
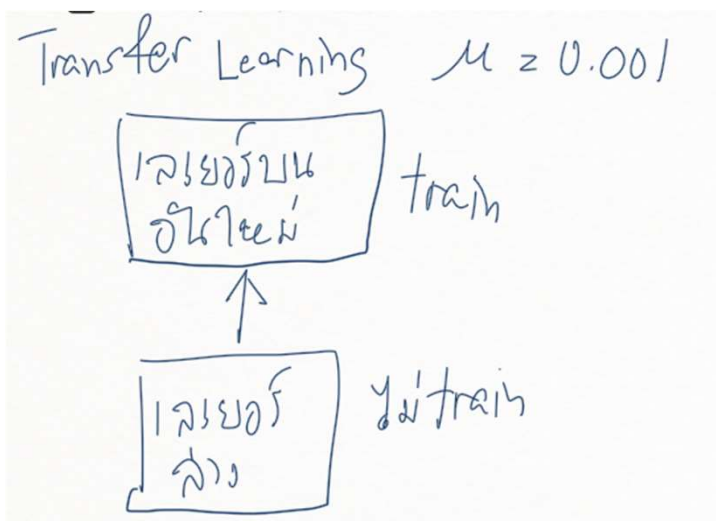
Transfer learning – fine tuning

- คือการนำ NN ขนาดใหญ่ที่เทรนบน dataset ImageNet (15M รูป) นำมาเทรนต่อบน dataset ใหม่

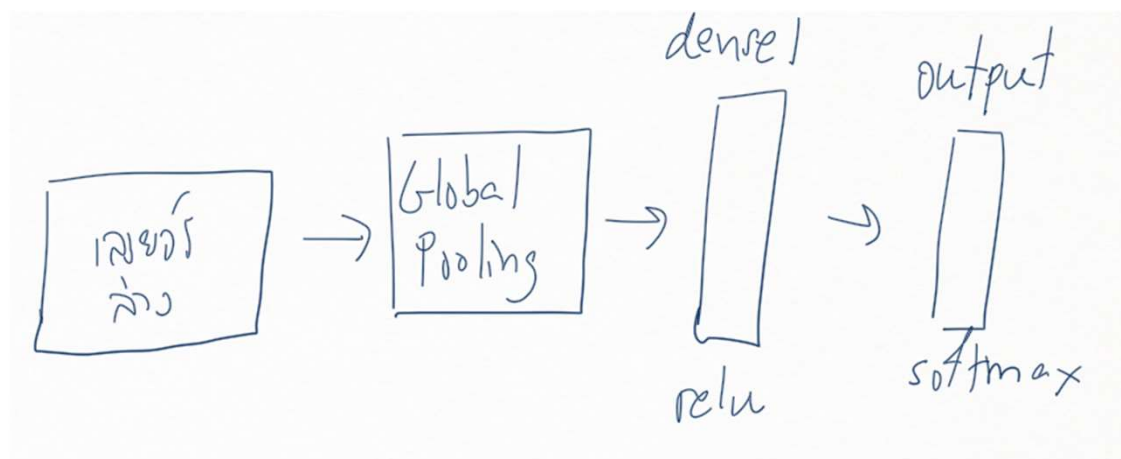


<https://cloud.google.com/tpu/docs/inception-v3-advanced>

Transfer learning – fine tuning



Transfer learning – fine tuning



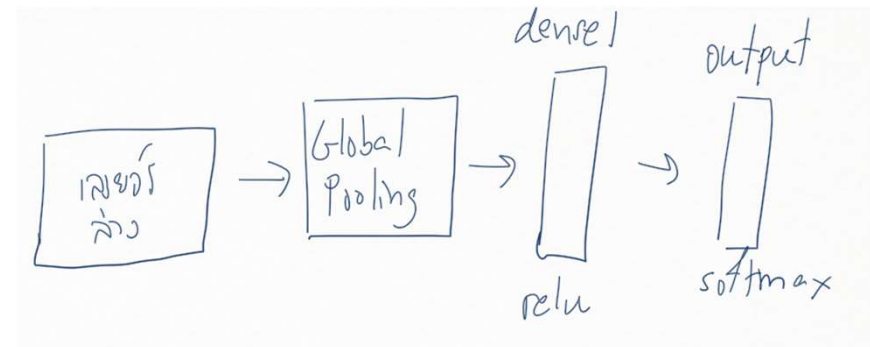
Keras.Applications

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
ResNeXt50	96 MB	0.777	0.938	25,097,128	-
ResNeXt101	170 MB	0.787	0.943	44,315,560	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset.

Example 05 – transfer learning

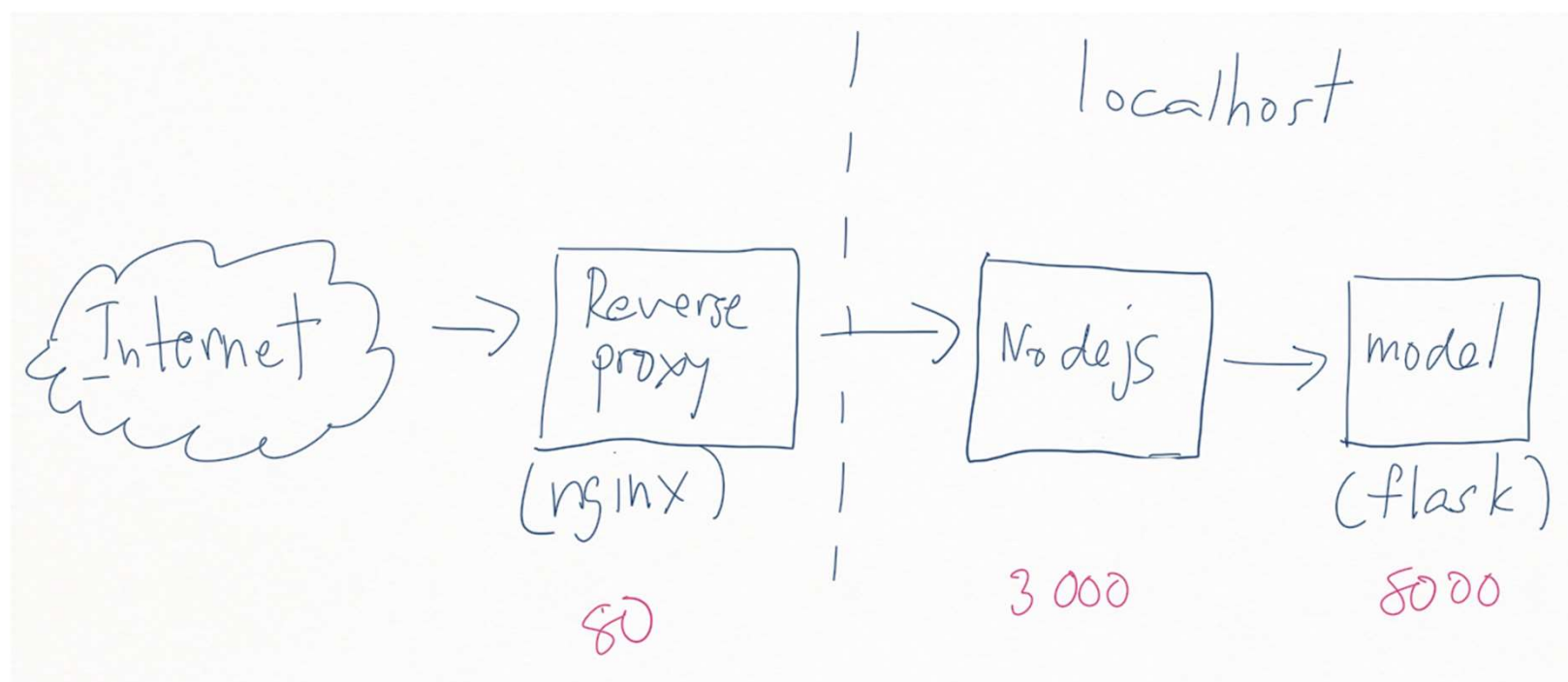
- ใช้ MobileNetV2
- flower dataset
- dense1 (100) -> dropout (0.5) -> softmax



การ deploy model ที่เทรนเสร็จแล้ว

1. Save model เป็น .h5
2. Setup Python (3.6)
 - `pip install tensorflow==1.13.1 numpy flask Pillow gunicorn`
3. Setup nodejs
 - Install npm packages: `npm install --save express express-formidable express-basic-auth request forever`
4. ก๊อปปี้ server.py, serverNode.js และไฟล์โมเดล .h5 ไว้ในโฟลเดอร์เดียวกัน
5. `gunicorn -w 2 --timeout 200 server:app`
6. `node serverNode.js`
7. ทำ reverse proxy ไปที่ localhost:3000

การ deploy model ที่เทรนเสร็จแล้ว



การทำให้รันค้างไว้เมื่อ logout จาก server

- *gunicorn -w 2 --daemon --timeout 200 server:app*
- *forever start serverNode.js*

Lab 04 – fine-tuning
