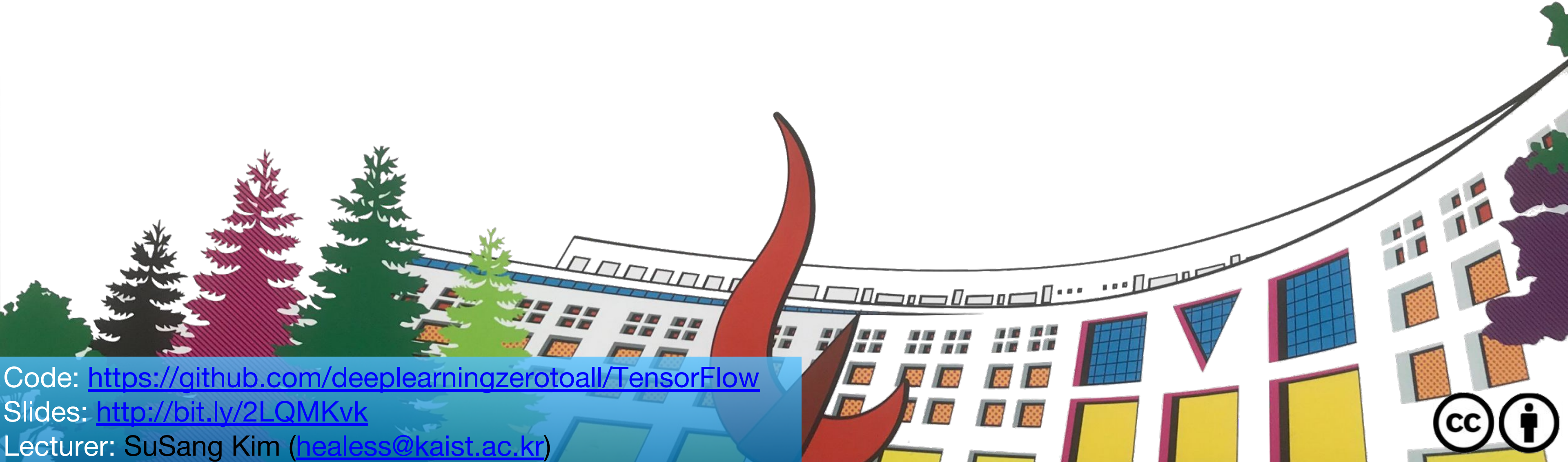
ML/DL for Everyone Season2

with TensorFlow

Lab 07-3 Application & Tips

Data & Learning



Code: https://github.com/deeplearningzerotoall/TensorFlow
Slides: http://bit.ly/2LQMKvk
Lecturer: SuSang Kim (healess@kaist.ac.kr)

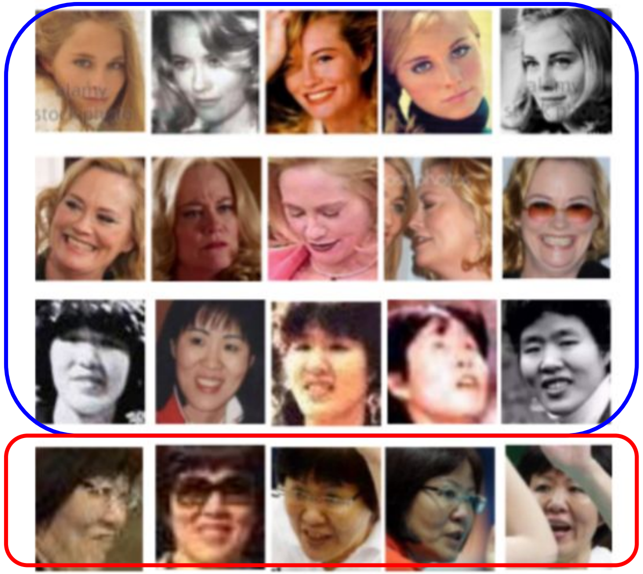
Application & Tips

- Data sets
  - Training / Validation / Testing
  - Evaluating a hypothesis
  - Anomaly Detection
- Learning
  - Online Learning vs Batch Learning
  - Fine tuning
  - Efficient Models
- Sample Data
  - Fashion MNIST / IMDB / CIFAR-100
- Summary

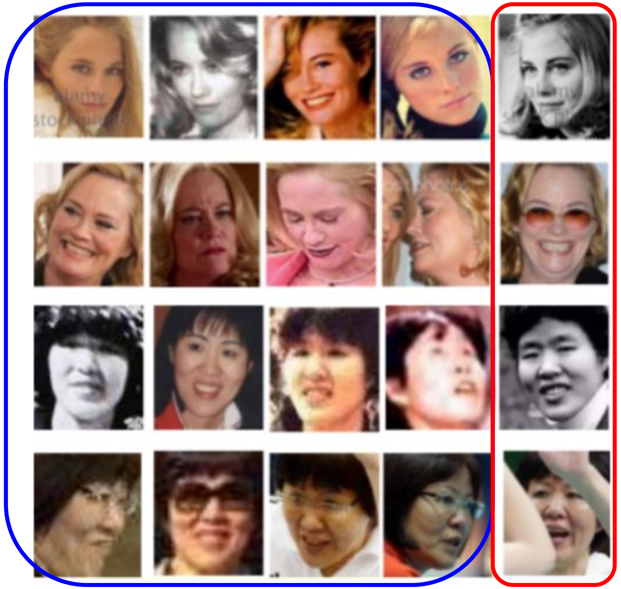
Andrew Ng's ML Class http://www.holehouse.org/mlclass/

Data sets

Training and Validation



VS

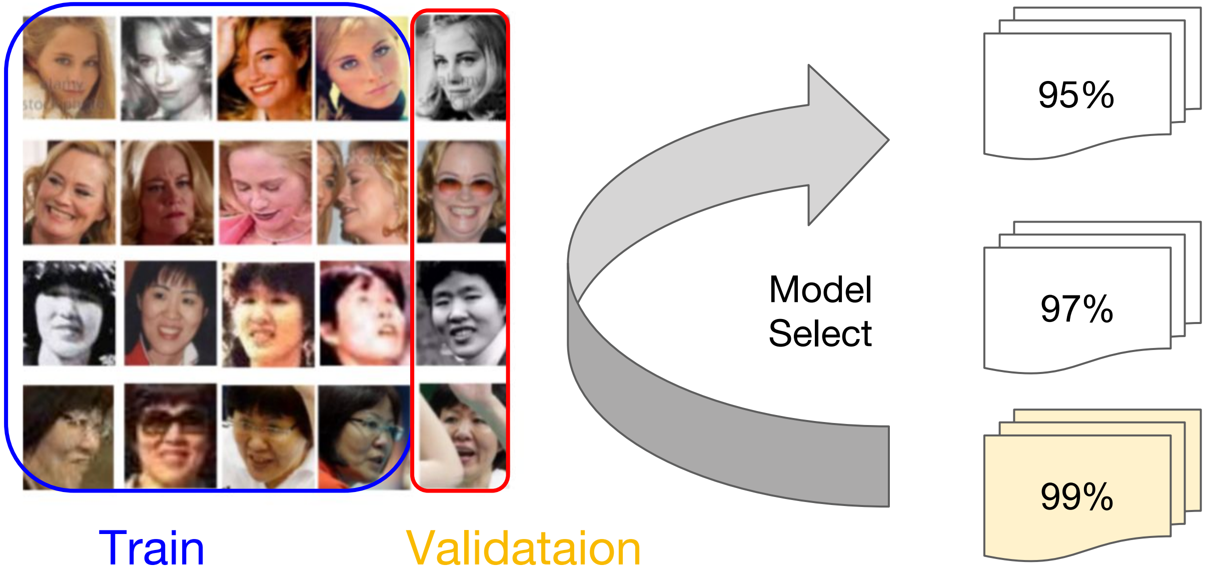


A set of example(label)

VGG Face2 : https://www.robots.ox.ac.uk/~vgg/publications/2018/Cao18/cao18.pdf

Data sets

Good Case



[Tensorflow Code]

```
mnist = tf.keras.datasets.mnist
(x_train, y_train),(x_test, y_test) = mnist.load_data() # 60,000 training / 10,000 testing images

model.fit(x_train, y_train, validation_split=0.2, epochs=5) # 20% Val data
```

Data sets

Evaluating a hypothesis

After fit parameters (select model)
Typical split might be 70:30 (training:test) for model testing

[Tensorflow Code]

```
test_acc = accuracy_fn(softmax_fn(x_test),y_test) # define hypothesis and test
model.evaluate(x_test, y_test) # Keras
```

CFP : http://www.cfpw.io/

Data sets

Anomaly detection

To be published in the proceedings of IPMI 2017

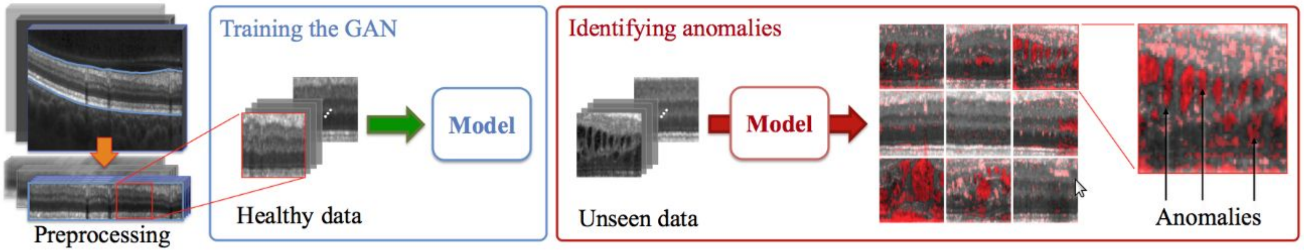


Fig. 1. Anomaly detection framework. The preprocessing step includes e flattening of the retinal area, patch extraction and intensity normalizatio adversarial training is performed on healthy data and testing is perfor unseen healthy cases and anomalous data.

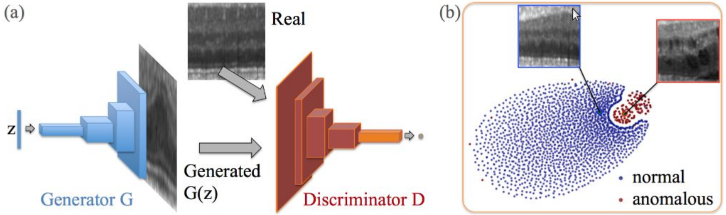


Fig. 2. (a) Deep convolutional generative adversarial network. (b) t-SNE embedding of normal (blue) and anomalous (red) images on the feature representation of the last convolution layer (orange in (a)) of the discriminator.

Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery : https://arxiv.org/pdf/1703.05921.pdf

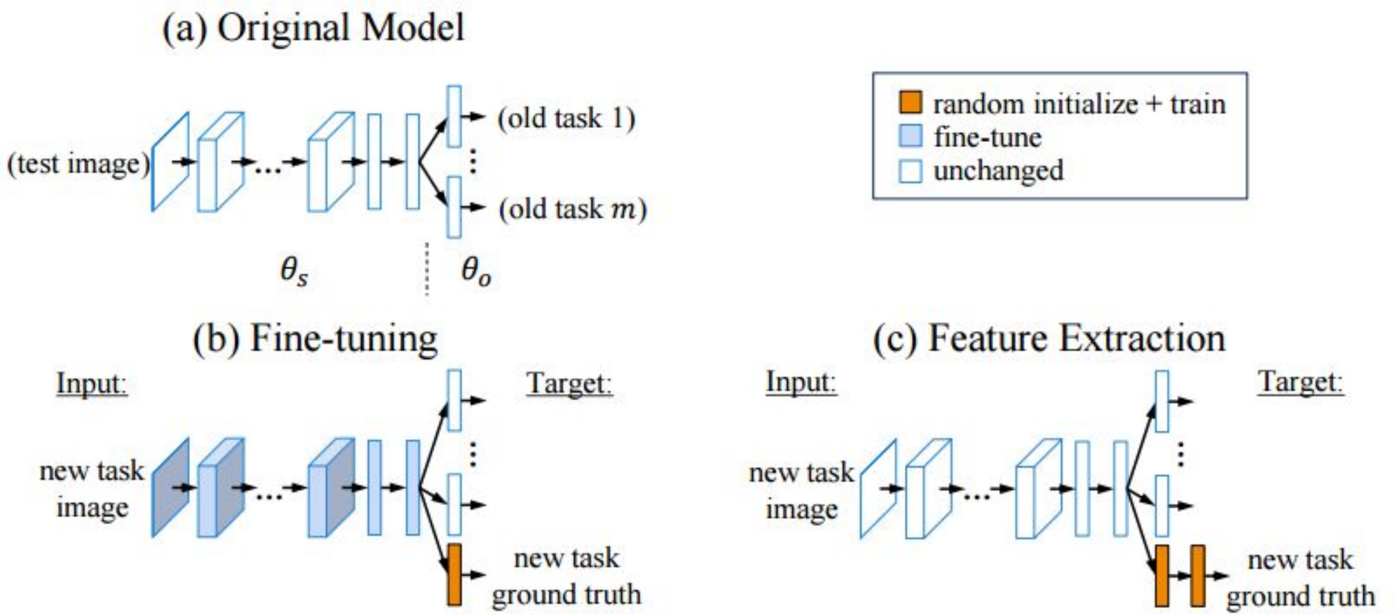
Learning

Online vs Batch

	Online Learning	Batch(Offline) Learning
Data	Fresh	Static
Network	connected	disconnected
Model	Updating	Static
Weight	Tunning	initialize
Infra(GPU)	Always	Per call
Application	Realtime Process	Stopping
Priority	Speed	Correctness

Learning

Fine Tuning / Feature Extraction



[Tensorflow Code]

```
saver = tf.train.import_meta_graph('my-model-1000.meta')
saver.restore(tf.train.latest_checkpoint('./'))
```

Learning without Forgetting : https://arxiv.org/pdf/1606.09282.pdf

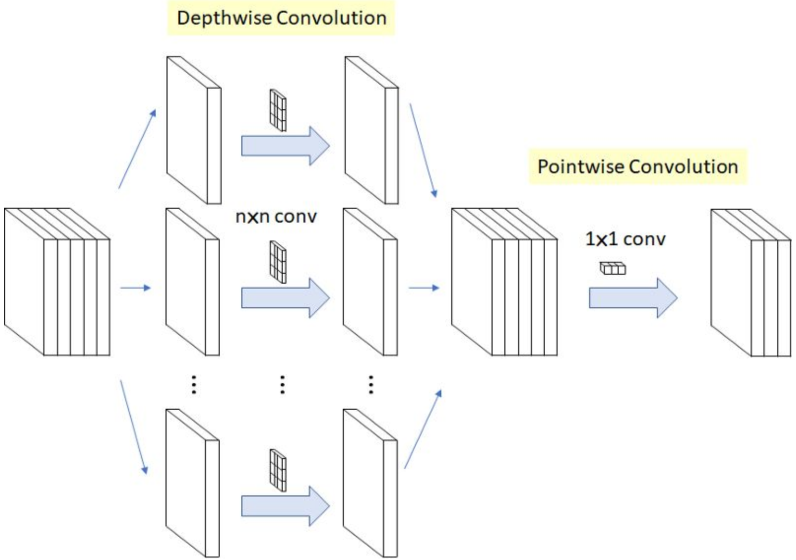
Fine tuning : https://goodtogreate.tistory.com/entry/Saving-and-Restoring

Learning

Efficient Models

Less inference time is needed,

So we need light weight
fully connected layers
really act as
1x1 convolutions
(Squeezenet, Mobilenet)



[Tensorflow Code]

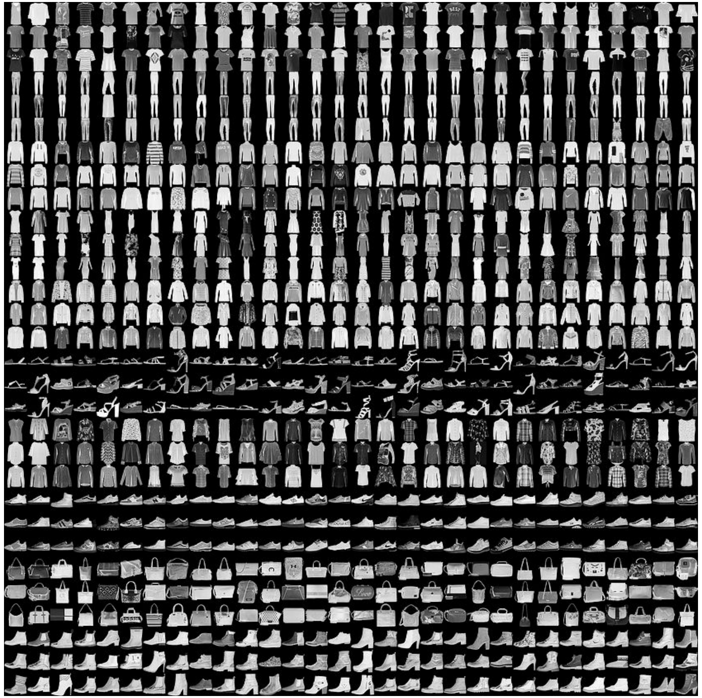
```
tf.nn.depthwise_conv2d(input, filter, strides, padding)
```

https://www.slideshare.net/healess/shufflenet-for-efficient-cnn
https://arstechnica.com/tech-policy/2018/05/police-use-of-amazons-face-recognition-service-draws-privacy-warnings/

Sample Data

Fashion MNIST-Image Classification

```
# Tensorflow Code
fashion_mnist = keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) =
fashion_mnist.load_data()

class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']

train_images = train_images / 255.0 # (60000, 28, 28)
test_images = test_images / 255.0 #(10000, 28, 28)

model = keras.Sequential([
   keras.layers.Flatten(input_shape=(28, 28)),
   keras.layers.Dense(128, activation=tf.nn.relu),
   keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
             loss='sparse_categorical_crossentropy',
             metrics=['accuracy'])

model.fit(train_images, train_labels, epochs=5)
test_loss, test_acc = model.evaluate(test_images, test_labels)
predictions = model.predict(test_images)
np.argmax(predictions[0]) # 9 label
```



https://github.com/zalandoresearch/fashion-mnist
https://www.tensorflow.org/tutorials/keras/basic_classification

Sample Data

IMDB-Text Classification

```
# Tensorflow Code
imdb = keras.datasets.imdb
(train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words=10000)
word_index = imdb.get_word_index()
# The first indices are reserved
word_index = {k:(v+3) for k,v in word_index.items()}
word_index["<PAD>"] = 0
word_index["<START>"] = 1
word_index["<UNK>"] = 2  # unknown
word_index["<UNUSED>"] = 3
reverse_word_index = dict([(value, key) for (key, value) in word_index.items()])
def decode_review(text):
   return ' '.join([reverse_word_index.get(i, '?') for i in text])

decode_review(train_data[4])
train_data = keras.preprocessing.sequence.pad_sequences(train_data,
value=word_index["<PAD>"], padding='post', maxlen=256)

vocab_size = 10000
model = keras.Sequential()
model.add(keras.layers.Embedding(vocab_size, 16))
model.add(keras.layers.GlobalAveragePooling1D())
model.add(keras.layers.Dense(16, activation=tf.nn.relu))
model.add(keras.layers.Dense(1, activation=tf.nn.sigmoid))
```

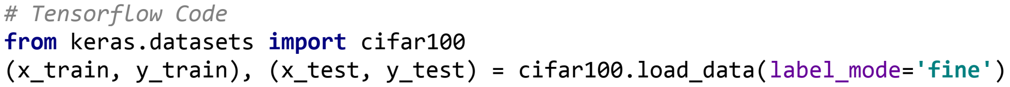
sentence	pos/neg
worst mistake of my life br br i picked this movie up at target for 5	0(neg)
this film was just brilliant casting location scenery story direction	1(pos)
this has to be one of the worst films of the 1990s	0(neg)

http://ai.stanford.edu/~amaas/data/sentiment/

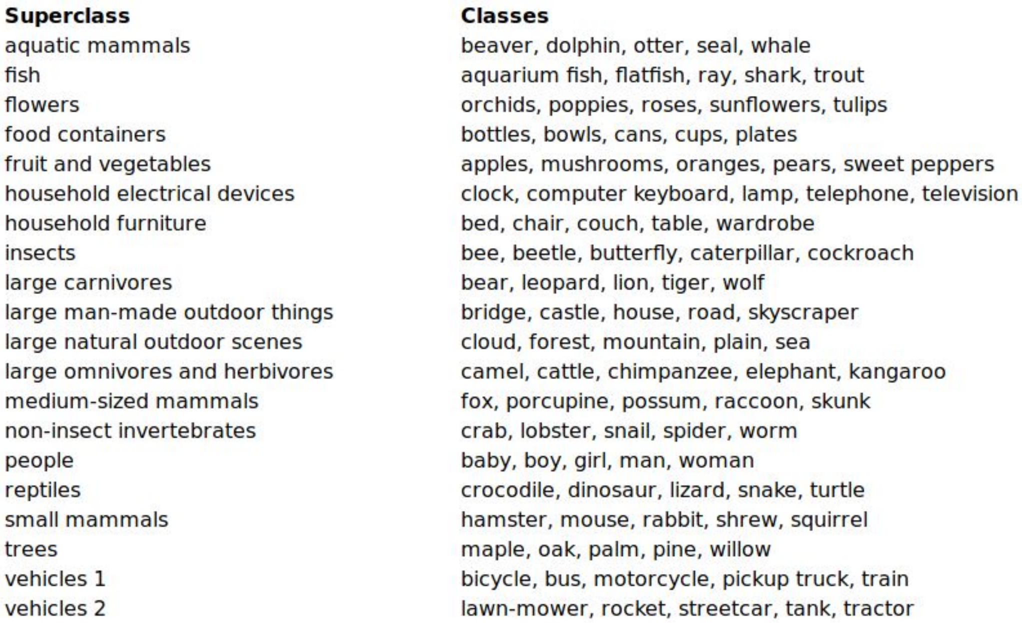
https://github.com/deeplearningzerotoall/TensorFlow/blob/master/lab-07-6-IMDB-introduction.ipynb

Sample Data

CIFAR-100



```
# Tensorflow Code
from keras.datasets import cifar100
(x_train, y_train), (x_test, y_test) = cifar100.load_data(label_mode='fine')
```



Superclass	Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

The CIFAR-100 dataset : https://www.cs.toronto.edu/~kriz/cifar.html

Summary

- Data sets
  - Training / Validation / Testing
  - Evaluating a hypothesis
  - Anomaly Detection
- Learning
  - Online Learning vs Batch Learning
  - Fine tuning
  - Efficient Models
- Sample Data
  - Fashion MNIST / IMDB / CIFAR-100