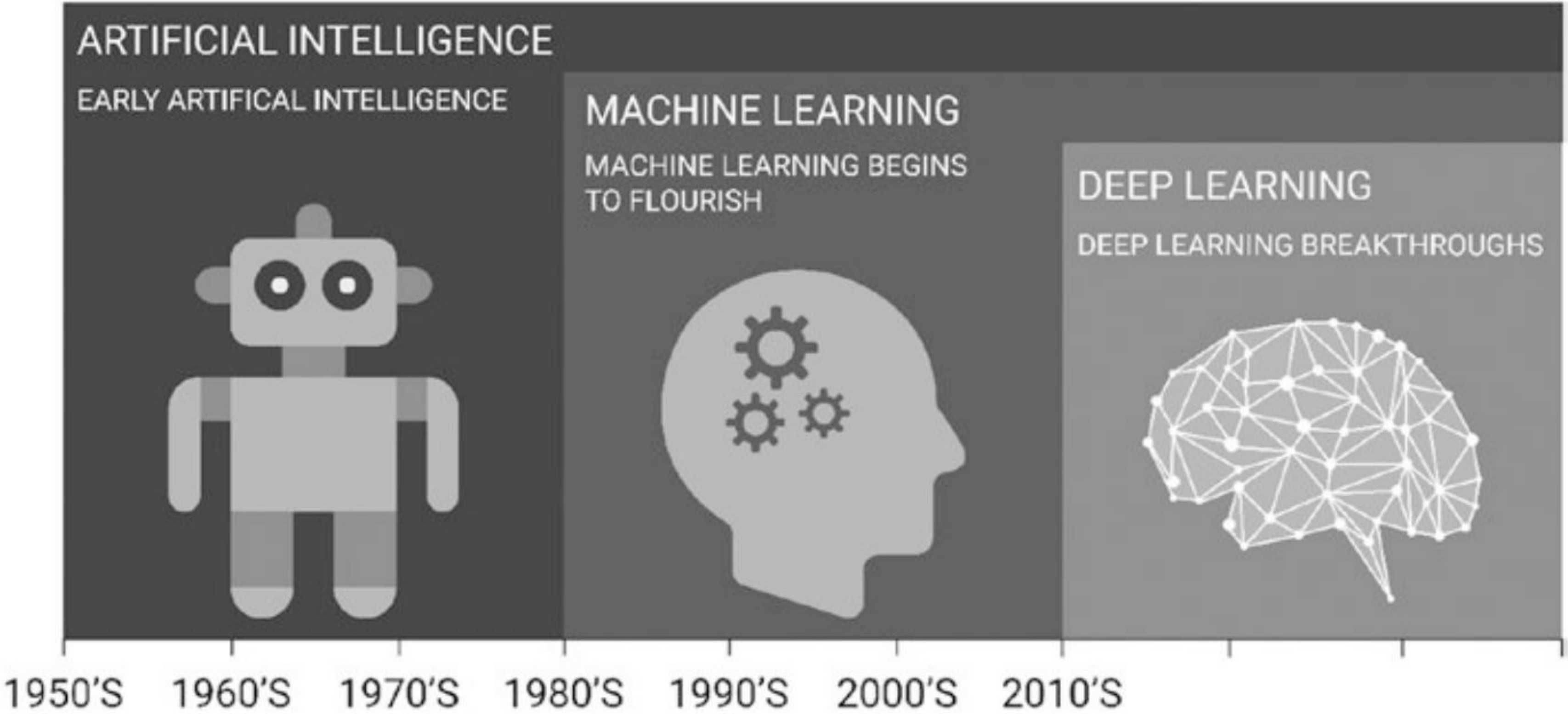
介紹AI深度學習

陽明大學 醫務管理研究所
陳 翎 助理教授

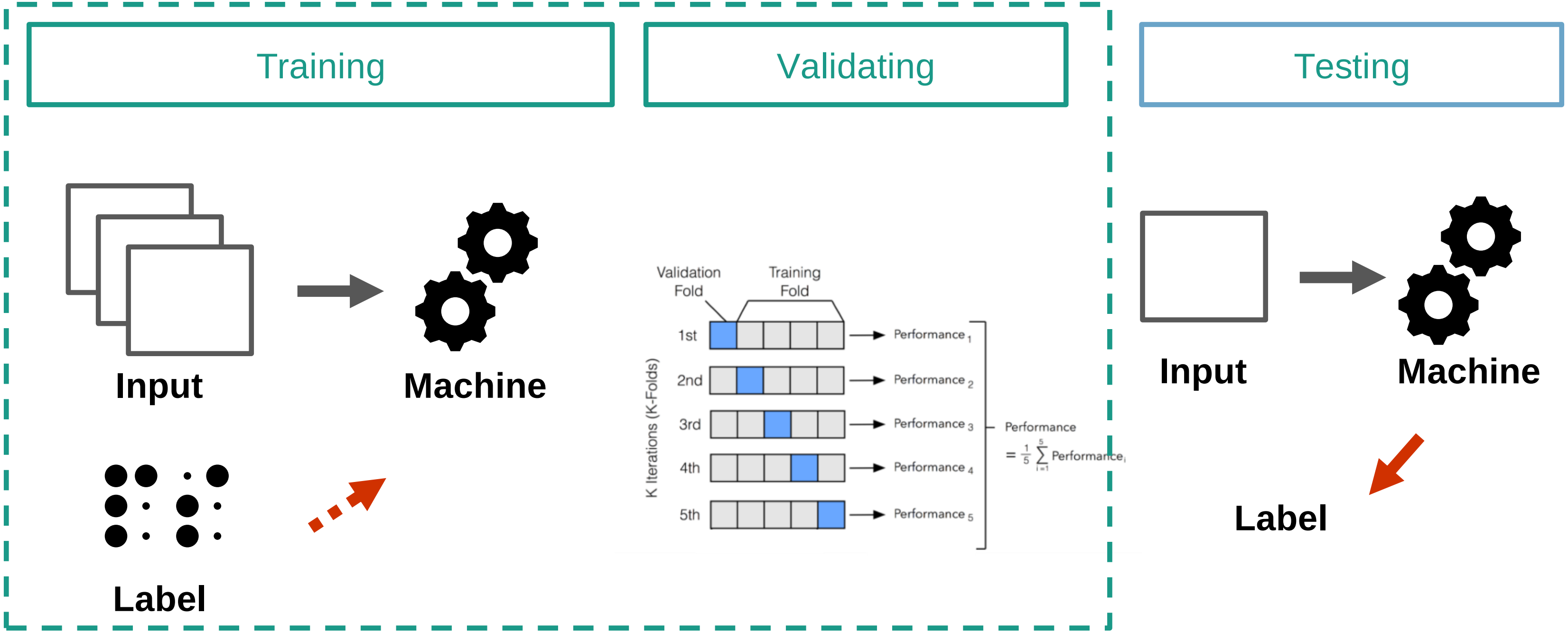
Machine Learning Recap

Concept Relationship



Panesar, A. (2019). Machine Learning and AI for Healthcare. In *Machine Learning and AI for Healthcare*. Ch 1.

Machine Learning Pipeline



Training

Validating

Testing

Input

Machine

Label

Input

Machine

Label

Cross-validation image: http://ethen8181.github.io/machine-learning/model_selection/model_selection.html

What exactly does a machine learn?

$$argmin_w E(||Goal - F(x, w)||)$$

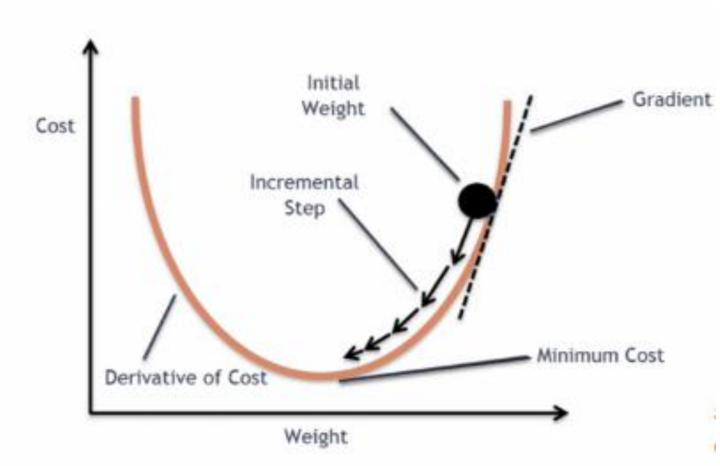


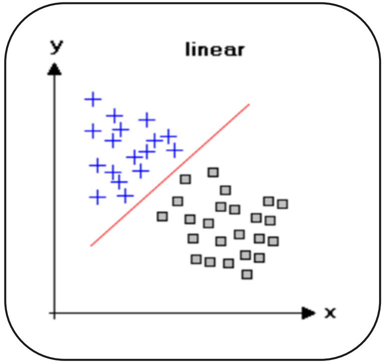
Image: https://blog.clairvoyantsoft.com/the-ascent-of-gradient-descent-23356390836f

What exactly does a machine learn?

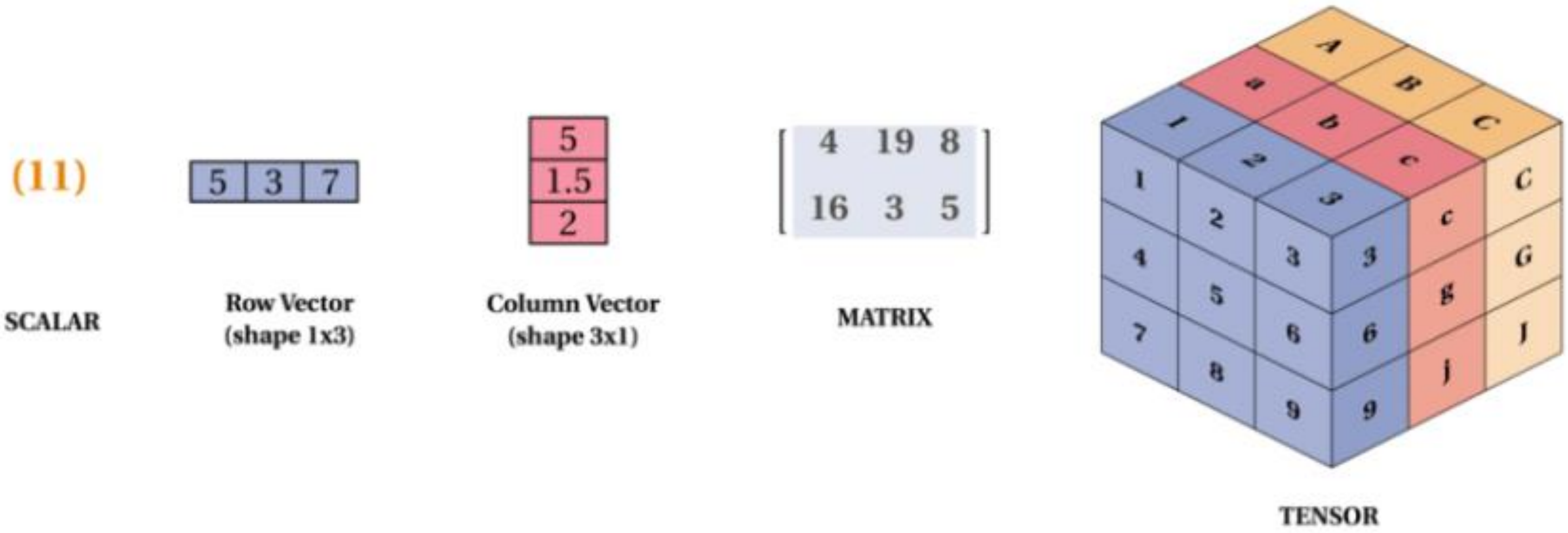
$$argmin_w E(||Goal - F(x, w)||)$$

- ➔ Example

$$MSE = \frac{1}{N} \sum_{i=1}^{n} (y_i - f(x_i))^2$$



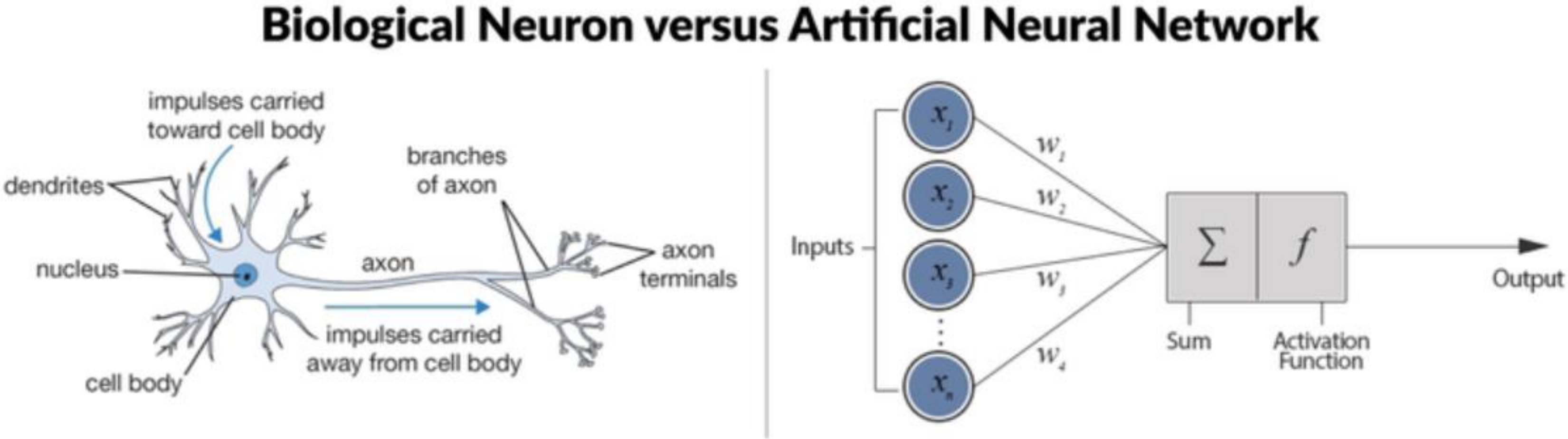
Vector, Matrix, Tensor



https://dev.to/mmithrakumar/scalars-vectors-matrices-and-tensors-with-tensorflow-2-0-1f66

Deep Learning Basics

Neural Networks

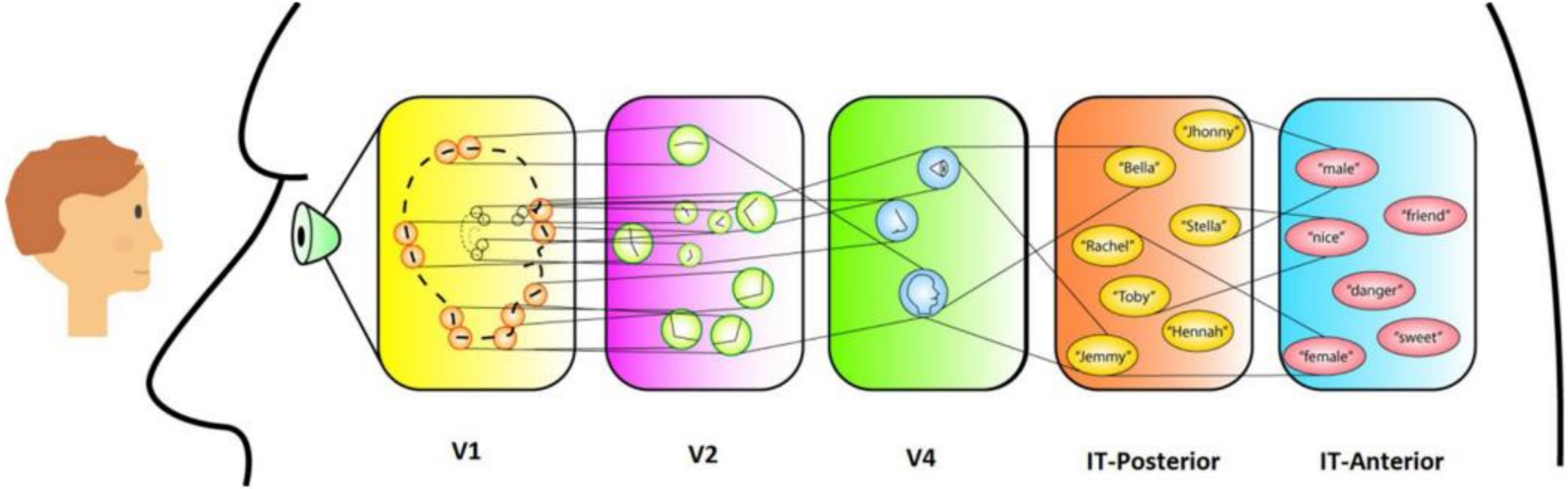

Biological Neuron versus Artificial Neural Network

$$y = f(\sum_{i=1}^{D} w_i * x_i)$$

$$y = f(w_1 * x_1 + w_2 * x_2 + \dots w_D * x_D)$$

https://www.analyticsvidhya.com/blog/2017/05/25-must-know-terms-concepts-for-beginners-in-deep-learning/

Deep learning: layers of abstraction



Saba, L., Biswas, M., Kuppili, V., Cuadrado Godia, E., Suri, H. S., Edla, D. R., Omerzu, T., Laird, J. R., Khanna, N. N., Mavrogeni, S., Protogerou, A., Sfikakis, P. P., Viswanathan, V., Kitas, G. D., Nicolaides, A., Gupta, A., & Suri, J. S. (2019). The present and future of deep learning in radiology. *European Journal of Radiology*, *114*(February), 14–24.

Basic Concepts for Training

Epoch

- One iteration over the entire dataset

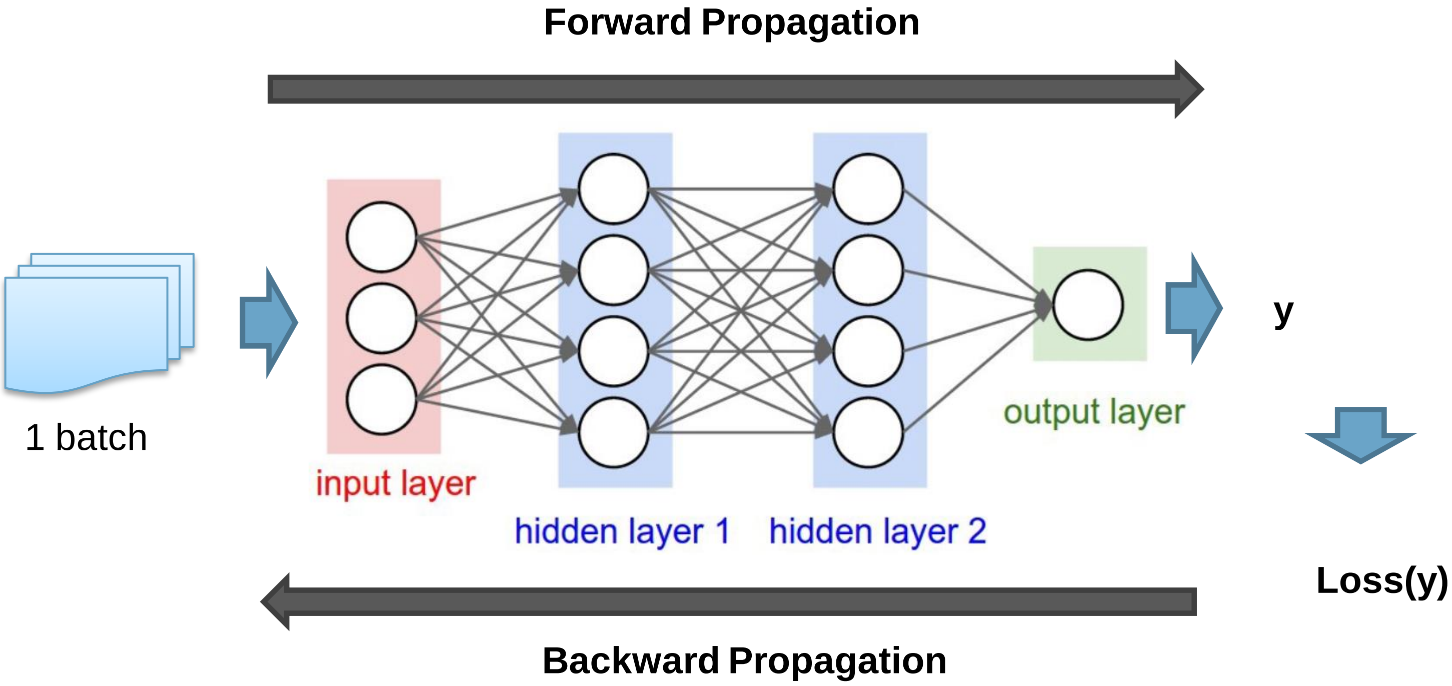
Batch

- The entire dataset is divided into a number of batches

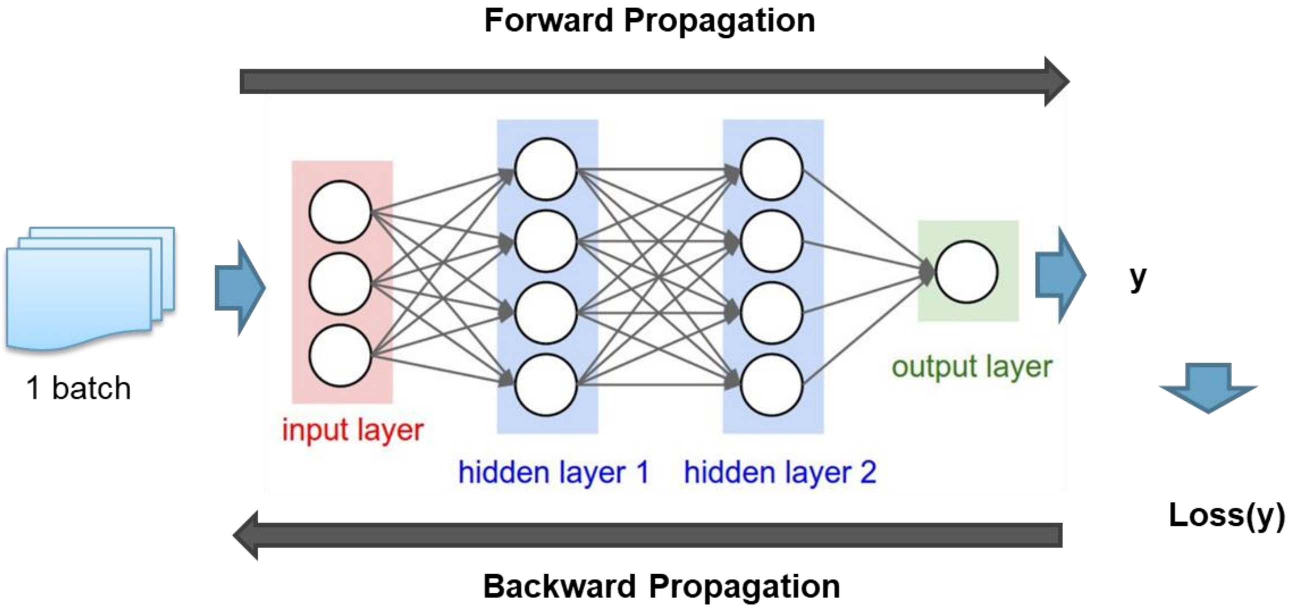
Iteration

- The number of batch runs over one epoch

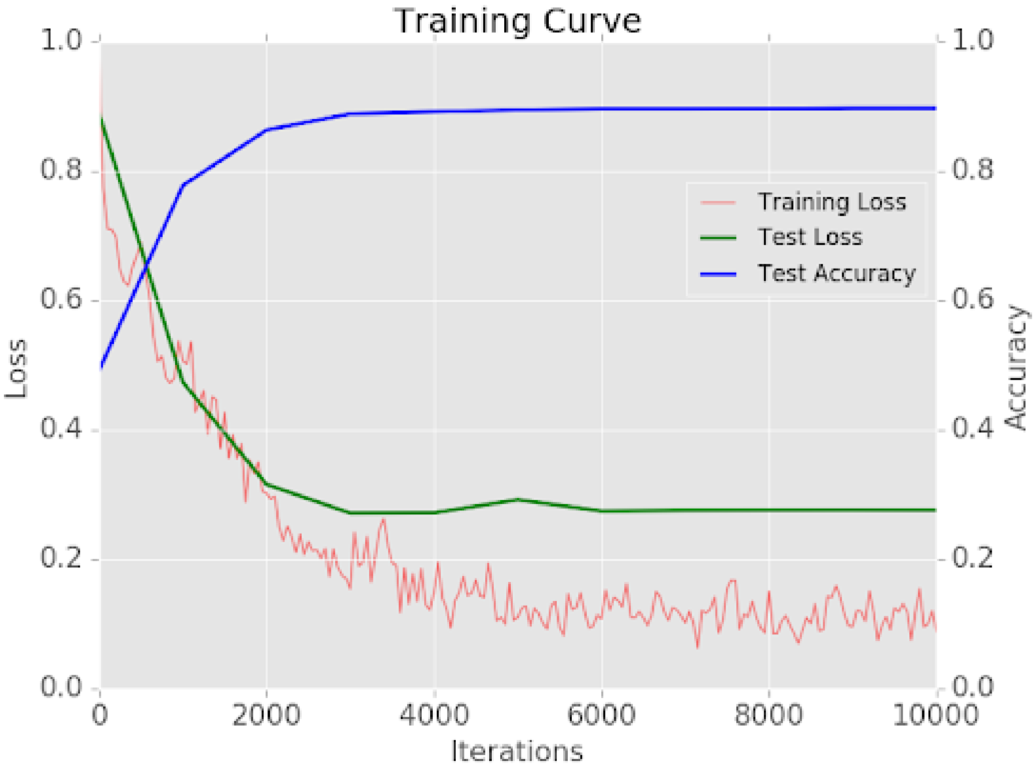
One iteration…



Batch size matters, why?

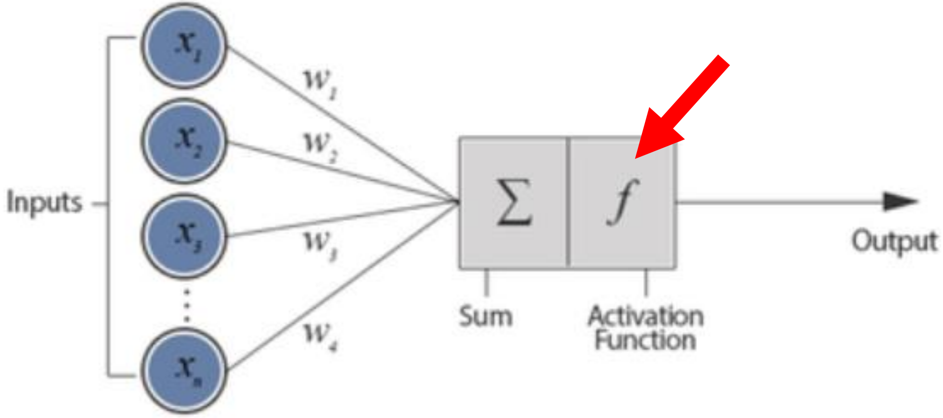


Training curve



Common Activation Functions

Activation Function



$$y = f(\sum_{i=1}^{D} w_i * x_i)$$

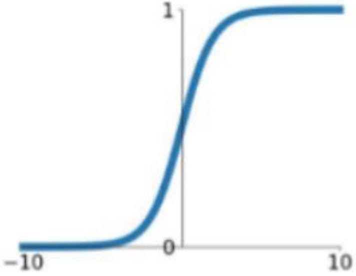
$$y = f(w_1 * x_1 + w_2 * x_2 + \ldots w_D * x_D)$$

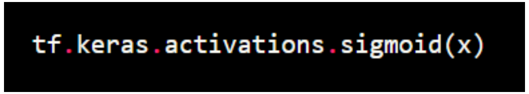
https://www.analyticsvidhya.com/blog/2017/05/25-must-know-terms-concepts-for-beginners-in-deep-learning/

Sigmoid

Sigmoid

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





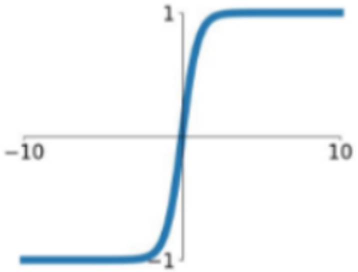
```
tf.keras.activations.sigmoid(x)
```

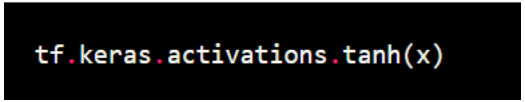
https://medium.com/hyunjulie/activation-functions-a-short-summary-8450c1b1d426

Tanh

tanh

$\tanh(x)$





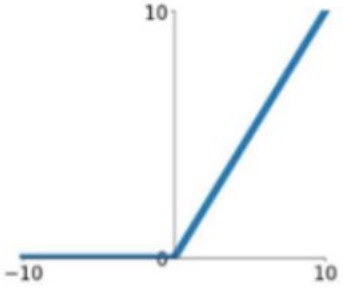
```
tf.keras.activations.tanh(x)
```

https://medium.com/hyunjulie/activation-functions-a-short-summary-8450c1b1d426

Rectified Linear Unit (ReLU)

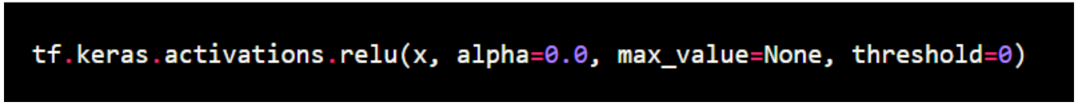
ReLU

$$\max(0, x)$$



Advantages:

- Handles vanishing gradient problem
- Non-saturating: not squashing real numbers to a range
- Faster convergence

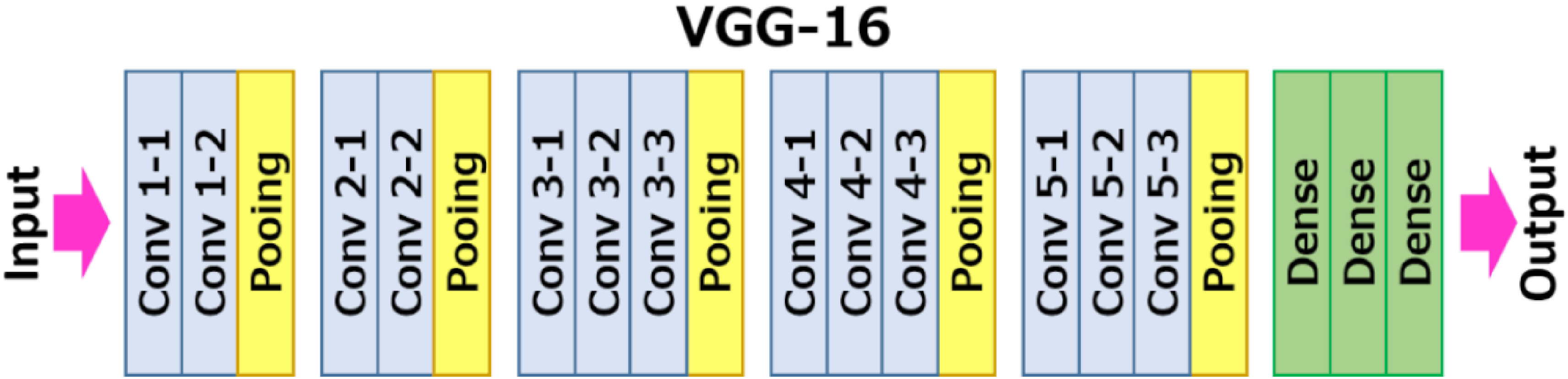


```
tf.keras.activations.relu(x, alpha=0.0, max_value=None, threshold=0)
```

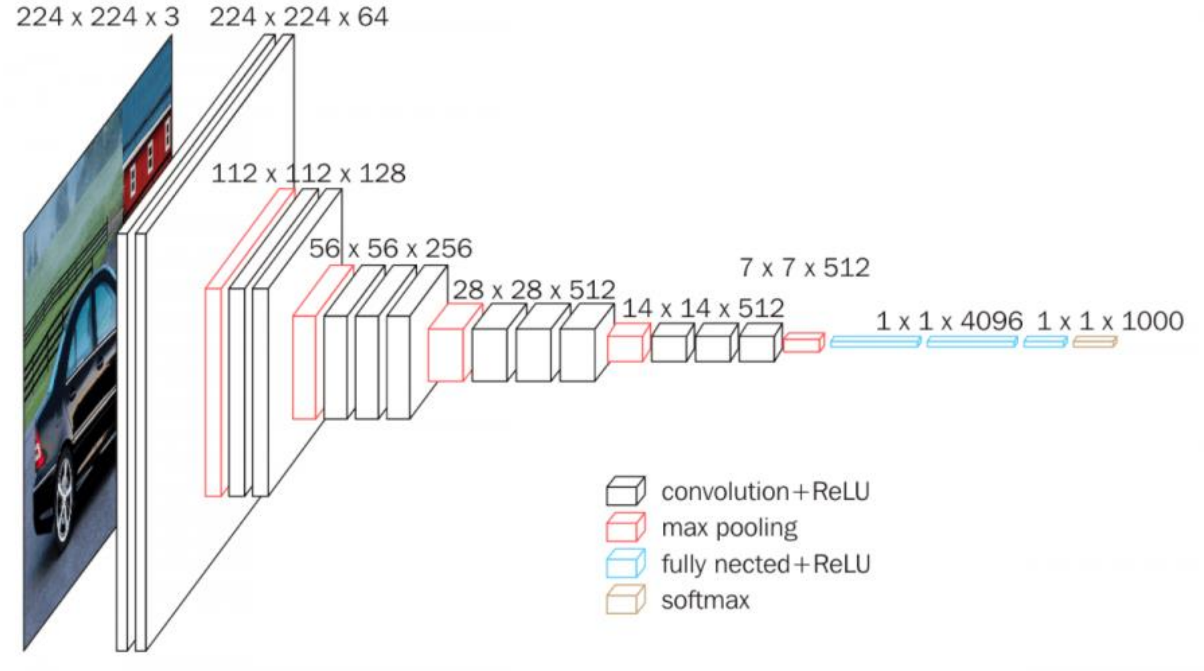
https://medium.com/hyunjulie/activation-functions-a-short-summary-8450c1b1d426

Commonly used layers

What does this network mean?

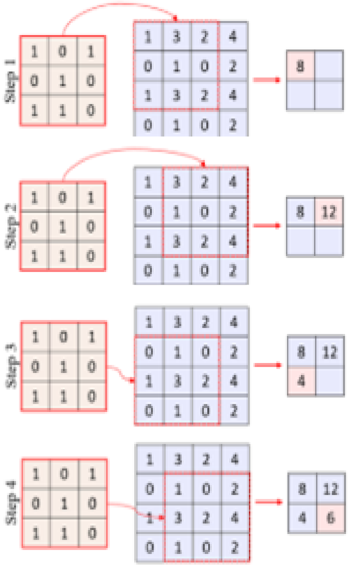


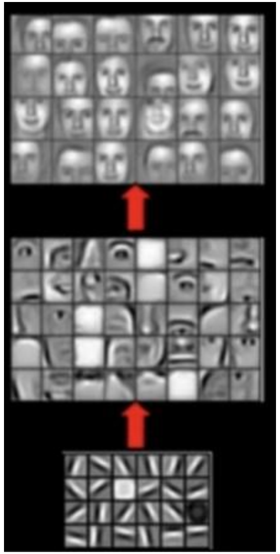
And this one?



https://neurohive.io/en/popular-networks/vgg16/

Convolution Layer





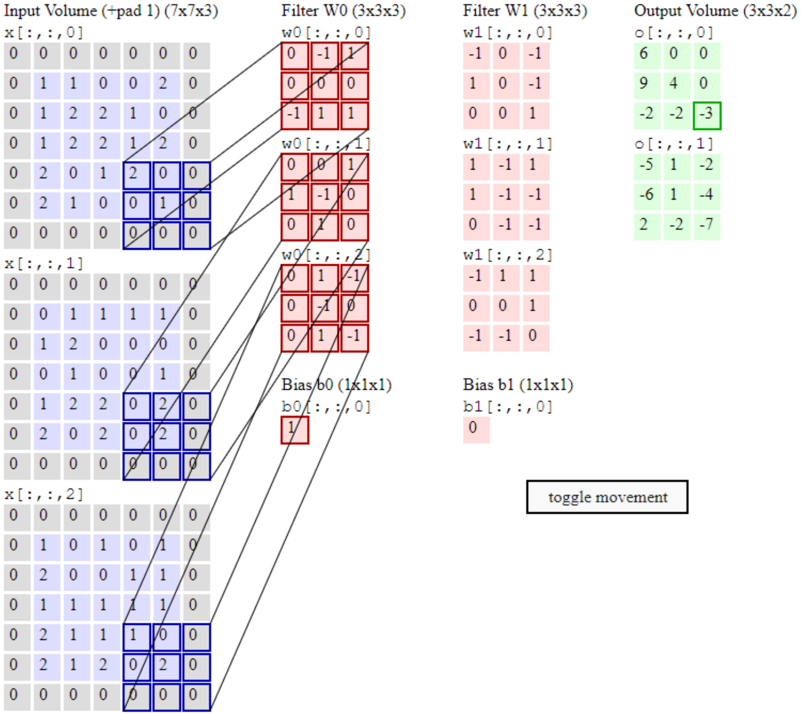
Output size = (W−F+2P)/S+1

W: input volume size
F: filter/channel size
P: the amount of zero padding used
S: the stride with which they are applied

This example:
(4-3+2*0)/1+1 = 2

https://www.topbots.com/advanced-topics-deep-convolutional-neural-networks/

Convolutional Layer



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	1	1	0	0	2	0
0	1	2	2	1	0	0
0	1	2	2	1	2	0
0	2	0	1	2	0	0
0	2	1	0	0	1	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	0	1	1	1	1	0
0	1	2	0	0	0	0
0	0	1	0	0	1	0
0	1	2	2	0	2	0
0	2	0	2	0	2	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	1	0	1	0	1	0
0	2	0	0	1	1	0
0	1	1	1	1	1	0
0	2	1	1	1	0	0
0	2	1	2	0	2	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

0	-1	1
0	0	0
-1	1	1

w0[:,:,1]

0	0	1
1	-1	0
0	1	0

w0[:,:,2]

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

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

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

w1[:,:,1]

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

w1[:,:,2]

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

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

o[:,:,0]

6	0	0
9	4	0
-2	-2	-3

o[:,:,1]

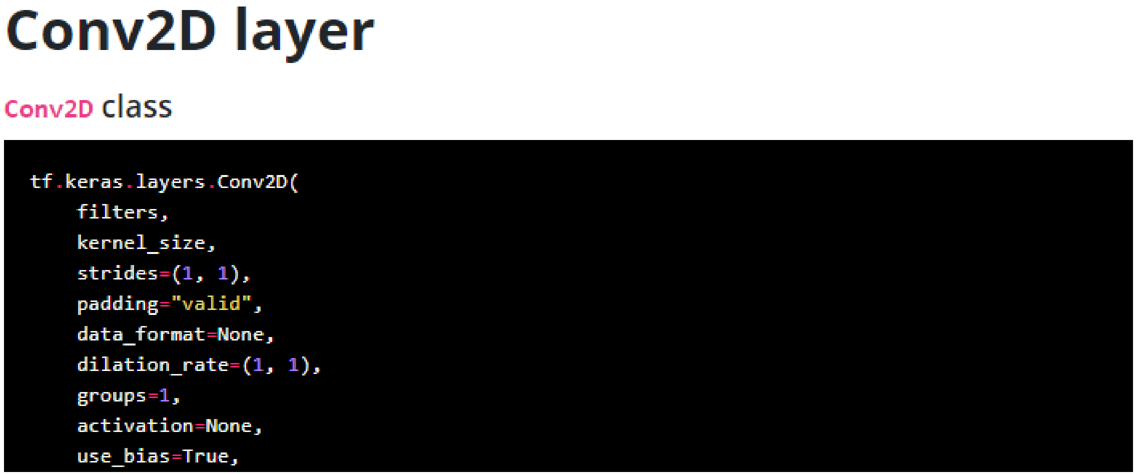
-5	1	-2
-6	1	-4
2	-2	-7

Output size = (W−F+2P)/S+1

This example:
(5-3+2*1)/2+1 = 3

https://cs231n.github.io/convolutional-networks/#conv

Convolutional Layer



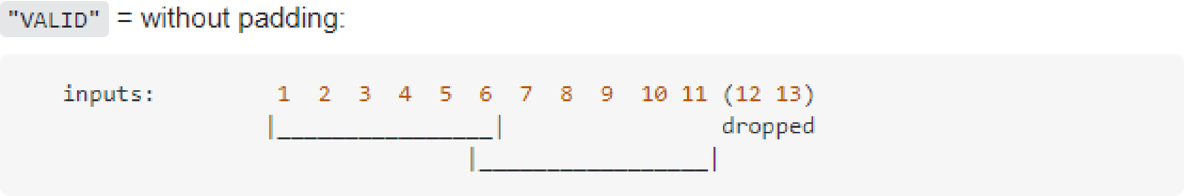
Conv2D layer

Conv2D class

```
tf.keras.layers.Conv2D(
    filters,
    kernel_size,
    strides=(1, 1),
    padding="valid",
    data_format=None,
    dilation_rate=(1, 1),
    groups=1,
    activation=None,
    use_bias=True,
```

https://keras.io/api/layers/convolution_layers/convolution2d/

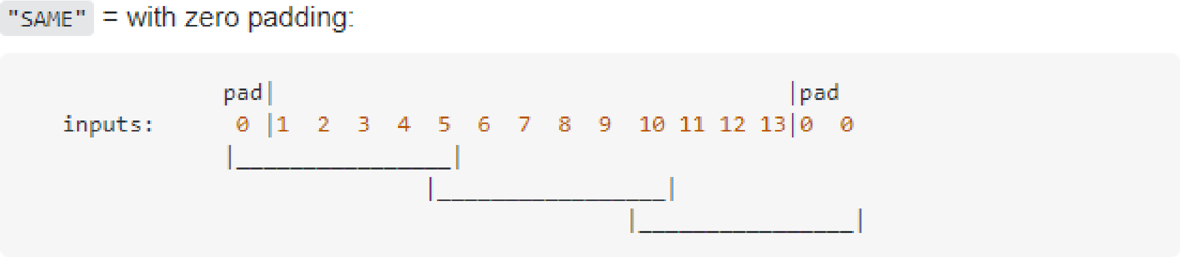
Padding



"VALID" = without padding:

```
inputs:         1  2  3  4  5  6  7  8  9  10 11 (12 13)
               |________________|                dropped
                              |_________________|
```

With "VALID" padding, there's no "made-up" padding inputs. The layer only uses **valid** input data.

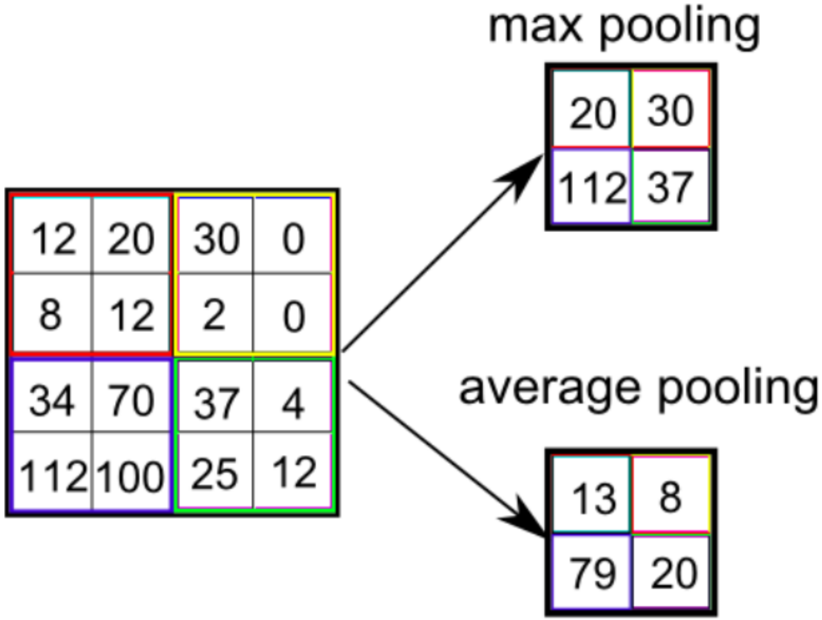


"SAME" = with zero padding:

```
            pad|                                      |pad
inputs:      0 |1  2  3  4  5  6  7  8  9  10 11 12 13|0  0
            |________________|
                           |_________________|
                                          |________________|
```

"SAME" tries to pad evenly left and right, but if the amount of columns to be added is odd, it will add the extra column to the right, as is the case in this example

Pooling Layer



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

20	30
112	37

average pooling

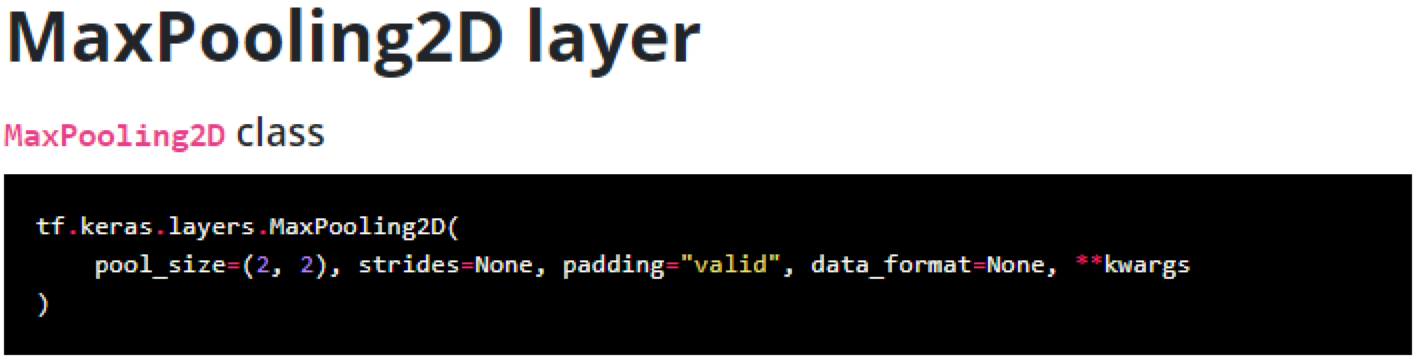
13	8
79	20

Output size = (W−F)/S+1

This example:
(4-2)/2+1 = 2

https://medium.com/analytics-vidhya/convolutional-neural-network-cnn-1bd606078aa5

Pooling Layer



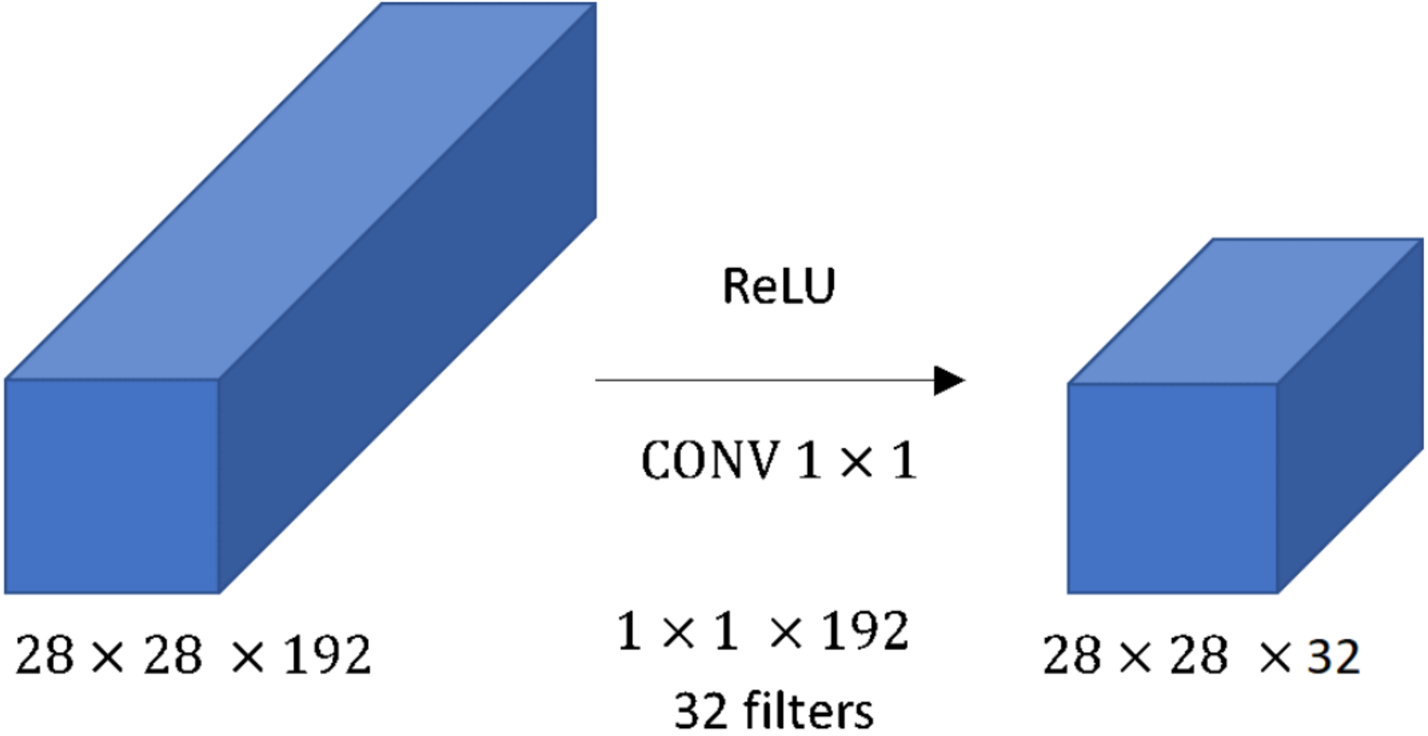
MaxPooling2D layer

MaxPooling2D class

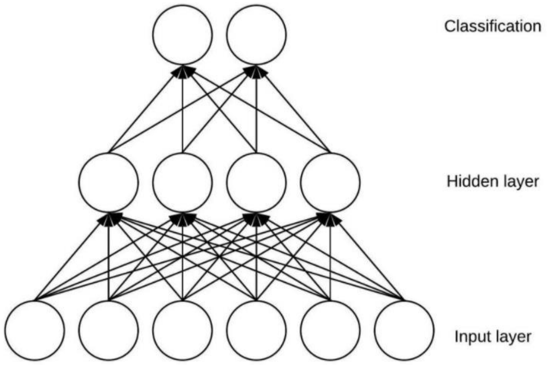
```
tf.keras.layers.MaxPooling2D(
    pool_size=(2, 2), strides=None, padding="valid", data_format=None, **kwargs
)
```

https://medium.com/analytics-vidhya/convolutional-neural-network-cnn-1bd606078aa5

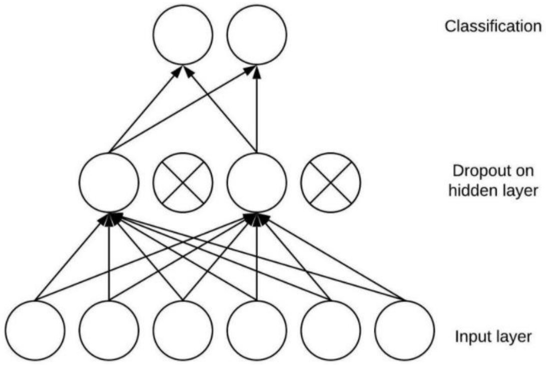
Using 1x1 Convolution



Dropout Layer



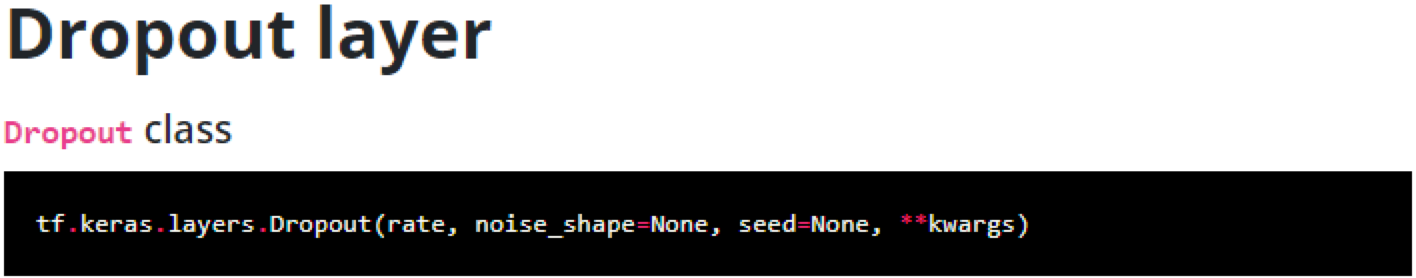
Without Dropout



With Dropout

https://www.baeldung.com/cs/ml-relu-dropout-layers

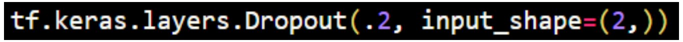
Dropout Layer



Dropout layer

Dropout class

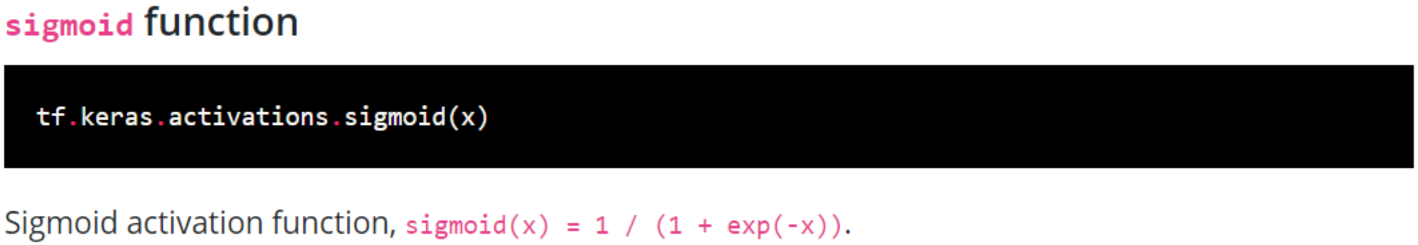
```
tf.keras.layers.Dropout(rate, noise_shape=None, seed=None, **kwargs)
```



```
tf.keras.layers.Dropout(.2, input_shape=(2,))
```

https://keras.io/api/layers/regularization_layers/dropout/

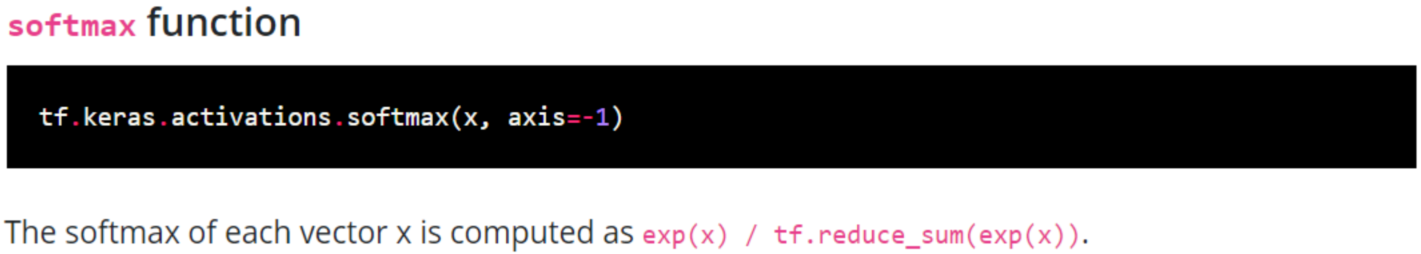
Output Layer



sigmoid function

```
tf.keras.activations.sigmoid(x)
```

Sigmoid activation function, `sigmoid(x) = 1 / (1 + exp(-x))`.



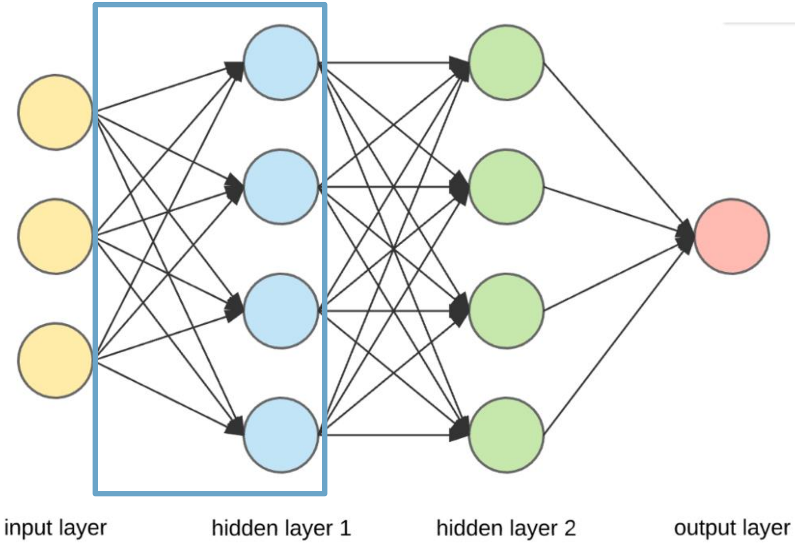
softmax function

```
tf.keras.activations.softmax(x, axis=-1)
```

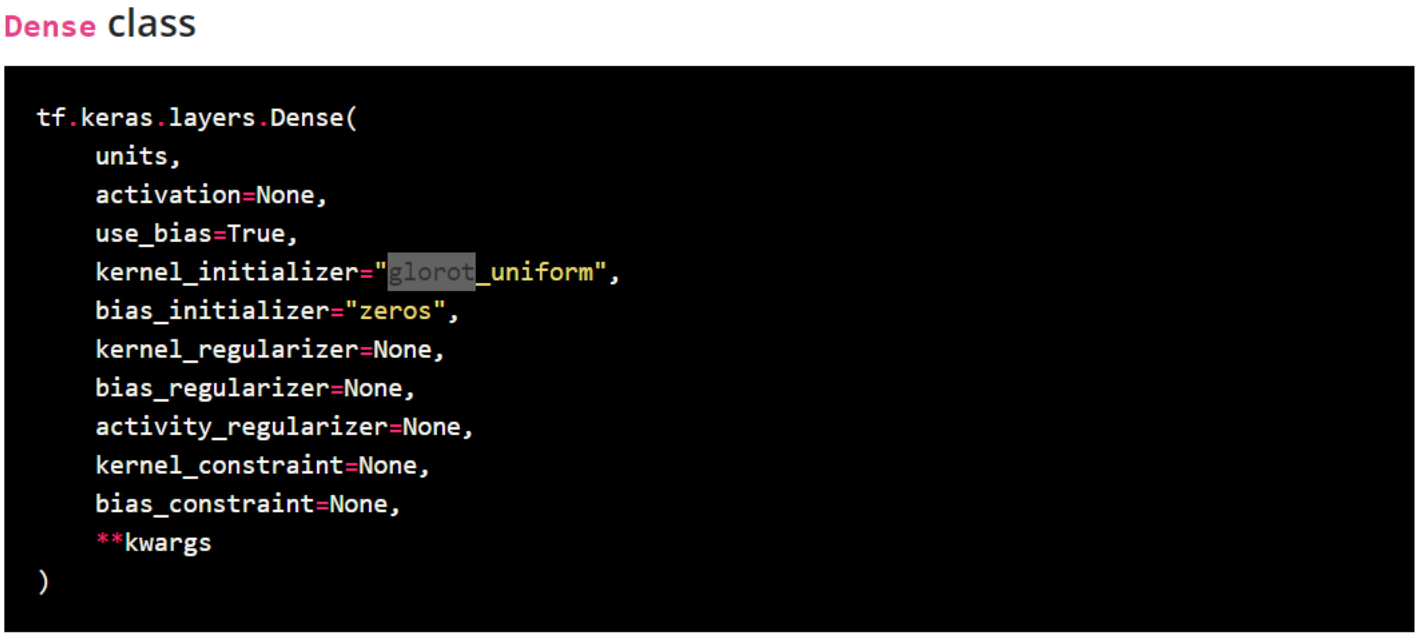
The softmax of each vector x is computed as `exp(x) / tf.reduce_sum(exp(x))`.

https://keras.io/api/layers

Dense/Fully Connected Layer



Dense/Fully Connected Layer

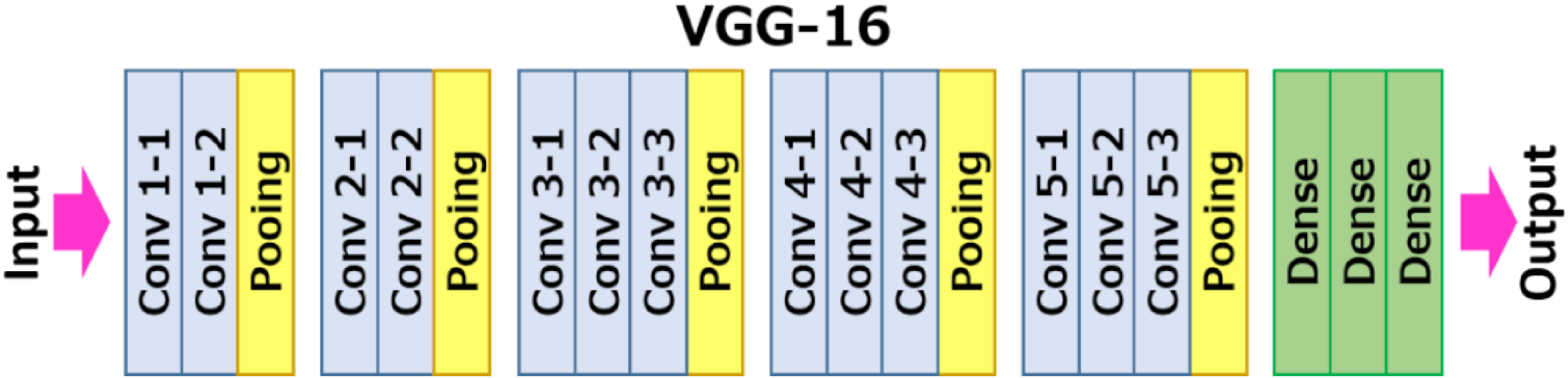


Dense class

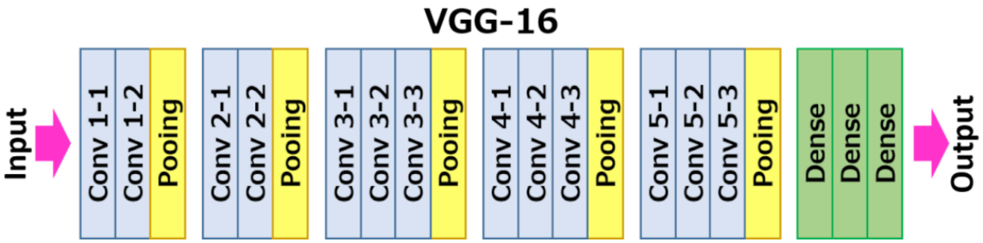
```
tf.keras.layers.Dense(
    units,
    activation=None,
    use_bias=True,
    kernel_initializer="glorot_uniform",
    bias_initializer="zeros",
    kernel_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    bias_constraint=None,
    **kwargs
)
```

https://keras.io/api/layers/core_layers/dense/

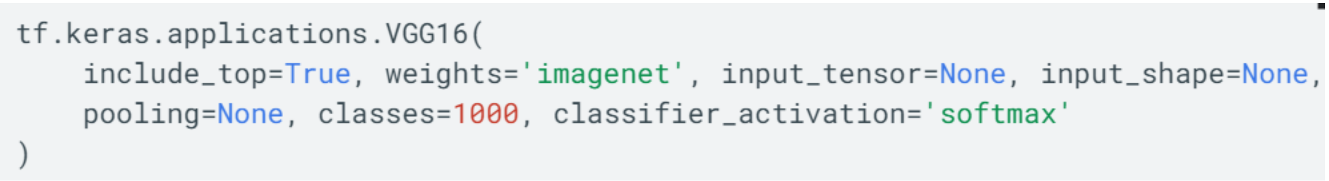
Now can you read this network?



A simple line of code…



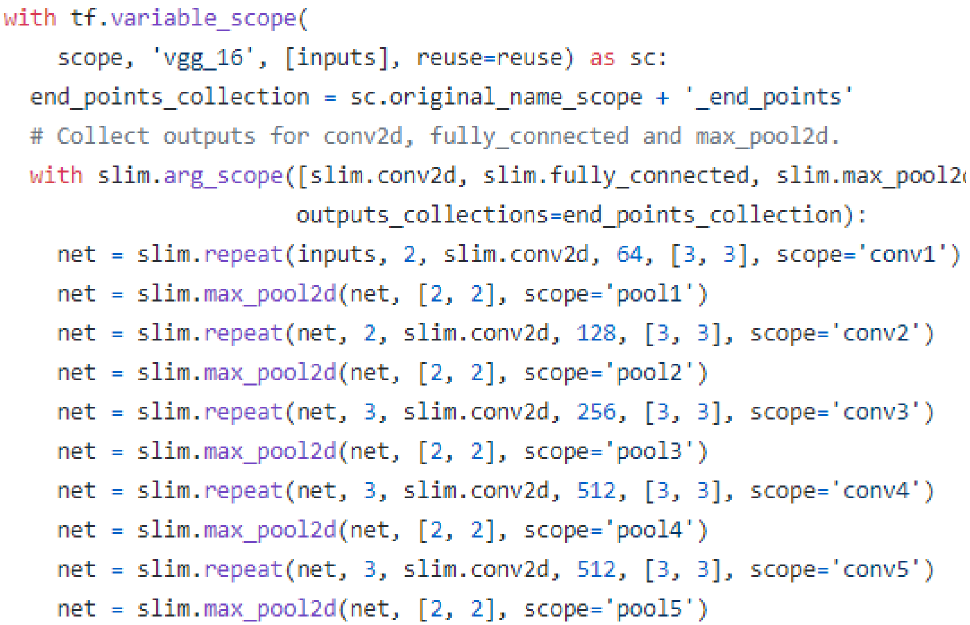
Keras TensorFlow



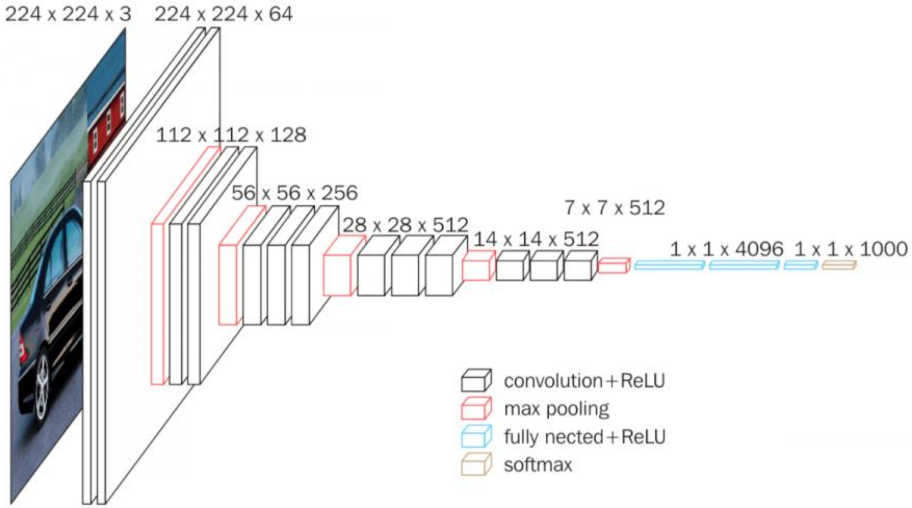
```
tf.keras.applications.VGG16(
    include_top=True, weights='imagenet', input_tensor=None, input_shape=None,
    pooling=None, classes=1000, classifier_activation='softmax'
)
```

https://www.tensorflow.org/api_docs/python/tf/keras/applications/VGG16

Behind the code…



```
with tf.variable_scope(
    scope, 'vgg_16', [inputs], reuse=reuse) as sc:
  end_points_collection = sc.original_name_scope + '_end_points'
  # Collect outputs for conv2d, fully_connected and max_pool2d.
  with slim.arg_scope([slim.conv2d, slim.fully_connected, slim.max_pool2
                      outputs_collections=end_points_collection):
    net = slim.repeat(inputs, 2, slim.conv2d, 64, [3, 3], scope='conv1')
    net = slim.max_pool2d(net, [2, 2], scope='pool1')
    net = slim.repeat(net, 2, slim.conv2d, 128, [3, 3], scope='conv2')
    net = slim.max_pool2d(net, [2, 2], scope='pool2')
    net = slim.repeat(net, 3, slim.conv2d, 256, [3, 3], scope='conv3')
    net = slim.max_pool2d(net, [2, 2], scope='pool3')
    net = slim.repeat(net, 3, slim.conv2d, 512, [3, 3], scope='conv4')
    net = slim.max_pool2d(net, [2, 2], scope='pool4')
    net = slim.repeat(net, 3, slim.conv2d, 512, [3, 3], scope='conv5')
    net = slim.max_pool2d(net, [2, 2], scope='pool5')
```



https://github.com/tensorflow/models/blob/master/research/slim/nets/vgg.py#L147; https://neurohive.io/en/popular-networks/vgg16/

Common Loss Functions

Learning is an Optimization Problem

Typically, a neural network model is trained using the **stochastic gradient descent** optimization algorithm and weights are updated using the **backpropagation**.

$$argmin_{w} E(\|Goal - F(x, w)\|)$$

“The function we want to minimize or maximize is called the objective function or criterion. When we are minimizing it, we may also call it the cost function, loss function, or error function.”

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, Issue 2). MIT press Cambridge.

Loss Function

- Regression

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|$$

Loss Function

- Classification – cross-entropy

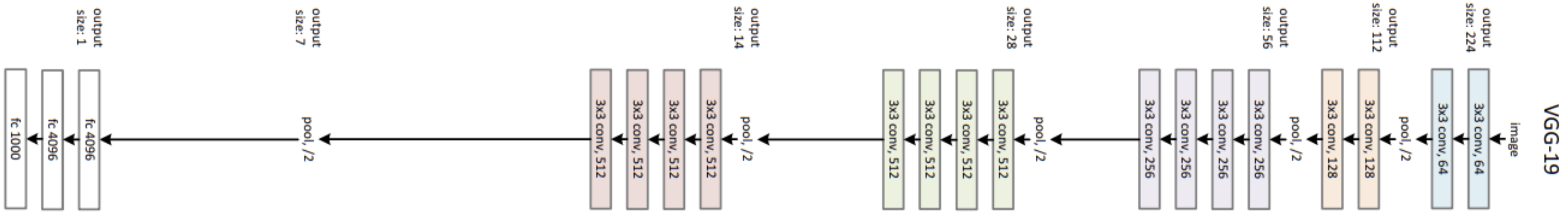
$$I(x) = -log_2(p(x))$$

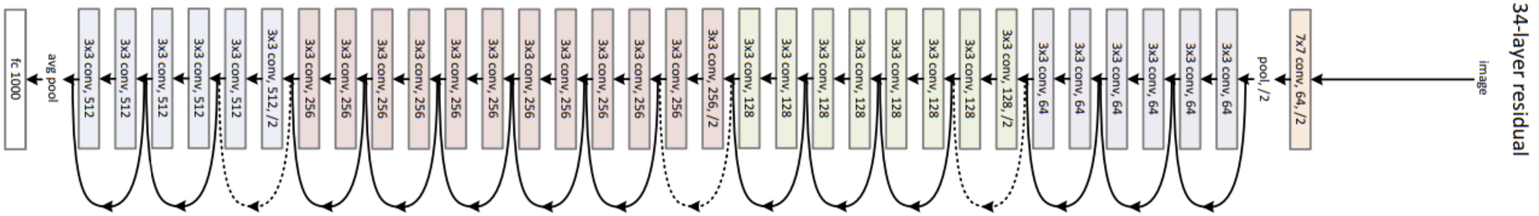
$$H = \sum_{c=1}^{C} \sum_{i=1}^{n} -y_{c,i} log_2(p_{c,i})$$

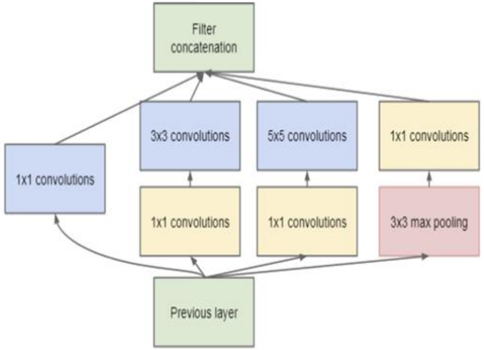
Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, Issue 2). MIT press Cambridge.

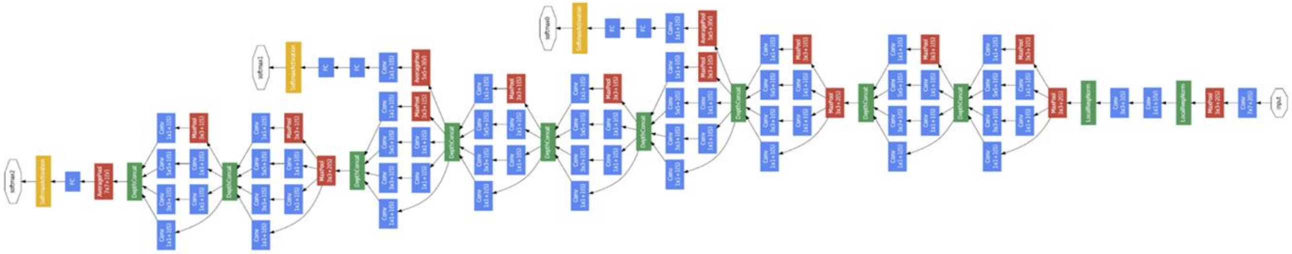
Standing on the shoulders of giants

Commonly used backbones









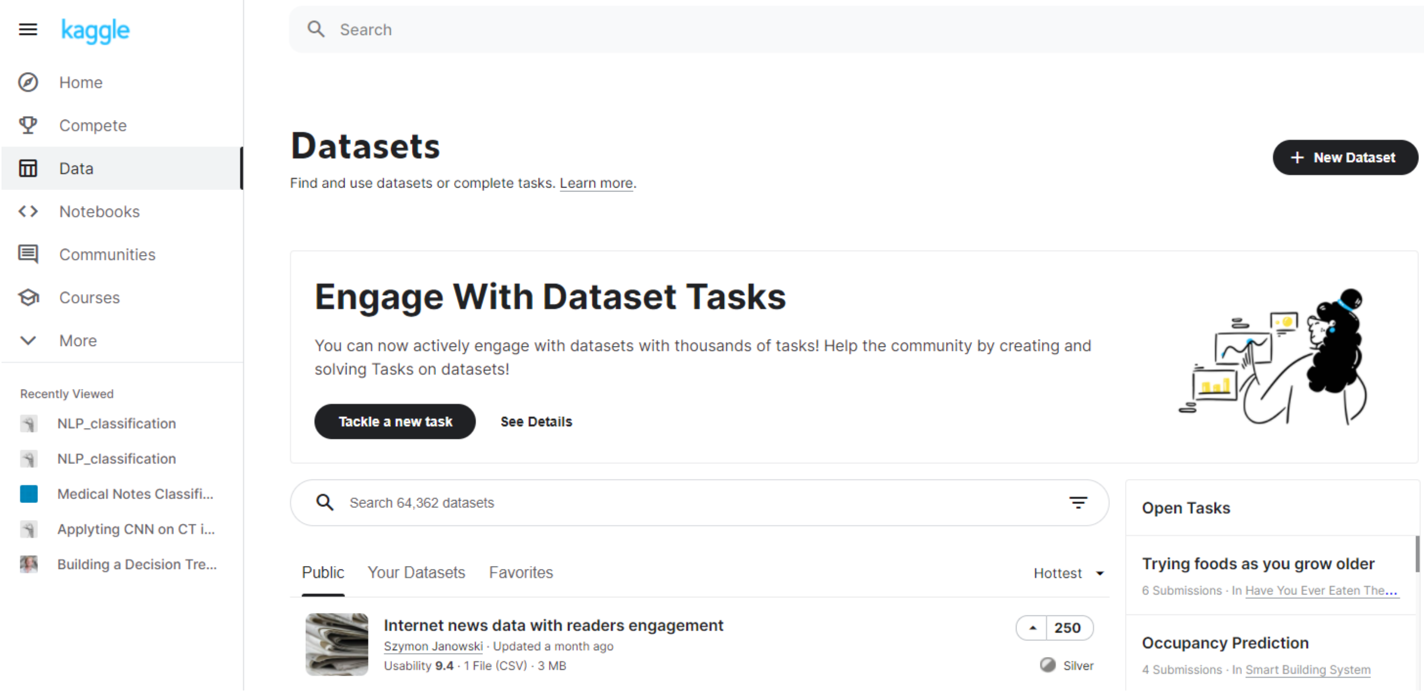
YOSHUA BENGIO,
GEOFFREY E. HINTON
AND YANN LECUN

For conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing

https://awards.acm.org/about/2018-turing

What's Next?

Kaggle



Recently Viewed

- NLP_classification
- NLP_classification
- Medical Notes Classifi...
- Applying CNN on CT i...
- Building a Decision Tre...

Datasets

Find and use datasets or complete tasks. Learn more.

Engage With Dataset Tasks

You can now actively engage with datasets with thousands of tasks! Help the community by creating and solving Tasks on datasets!

Search 64,362 datasets

Public | Your Datasets | Favorites | Hottest

Internet news data with readers engagement 250
Szymon Janowski · Updated a month ago
Usability 9.4 · 1 File (CSV) · 3 MB
Silver

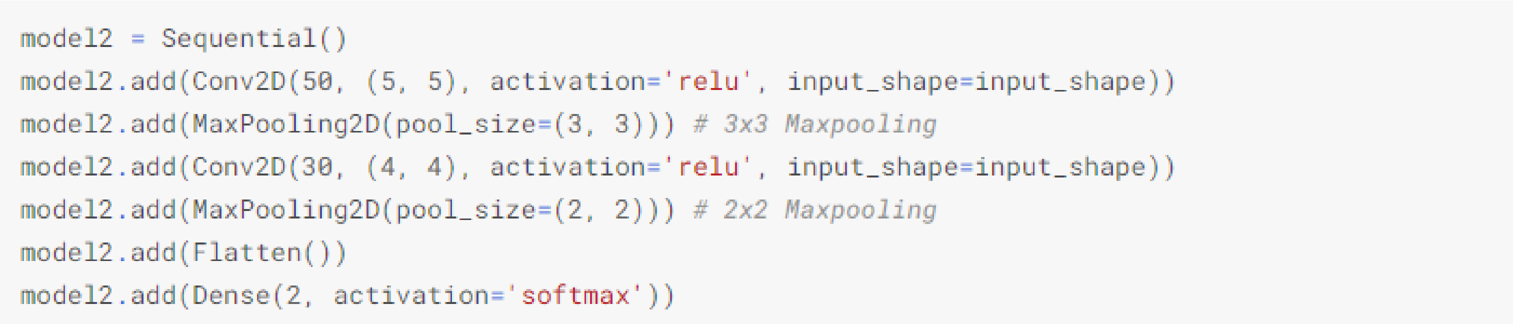
Open Tasks

Trying foods as you grow older
6 Submissions · In Have You Ever Eaten The...

Occupancy Prediction
4 Submissions · In Smart Building System

https://www.kaggle.com/

Example CNN



```
model2 = Sequential()
model2.add(Conv2D(50, (5, 5), activation='relu', input_shape=input_shape))
model2.add(MaxPooling2D(pool_size=(3, 3))) # 3x3 Maxpooling
model2.add(Conv2D(30, (4, 4), activation='relu', input_shape=input_shape))
model2.add(MaxPooling2D(pool_size=(2, 2))) # 2x2 Maxpooling
model2.add(Flatten())
model2.add(Dense(2, activation='softmax'))
```

https://www.kaggle.com/

Thank You ☺

請多多指教!