Deep Learning Fundamentals

An Interactive Introduction to Artificial Neural Networks and TensorFlow

POLL

Where are you?

- The Americas
- Europe / Middle East / Africa
- Asia-Pacific
- Extra-Terrestrial Space

POLL

What are you?

- Developer / Engineer
- Scientist / Analyst / Statistician / Mathematician
- Other

POLL

What's your level of experience with Deep Learning?

- Minimal
- Some theory
- Some theory and code
- Lots of theory and code

POLL

Use of accompanying Jupyter notebooks for interactive demos and in-training exercises?

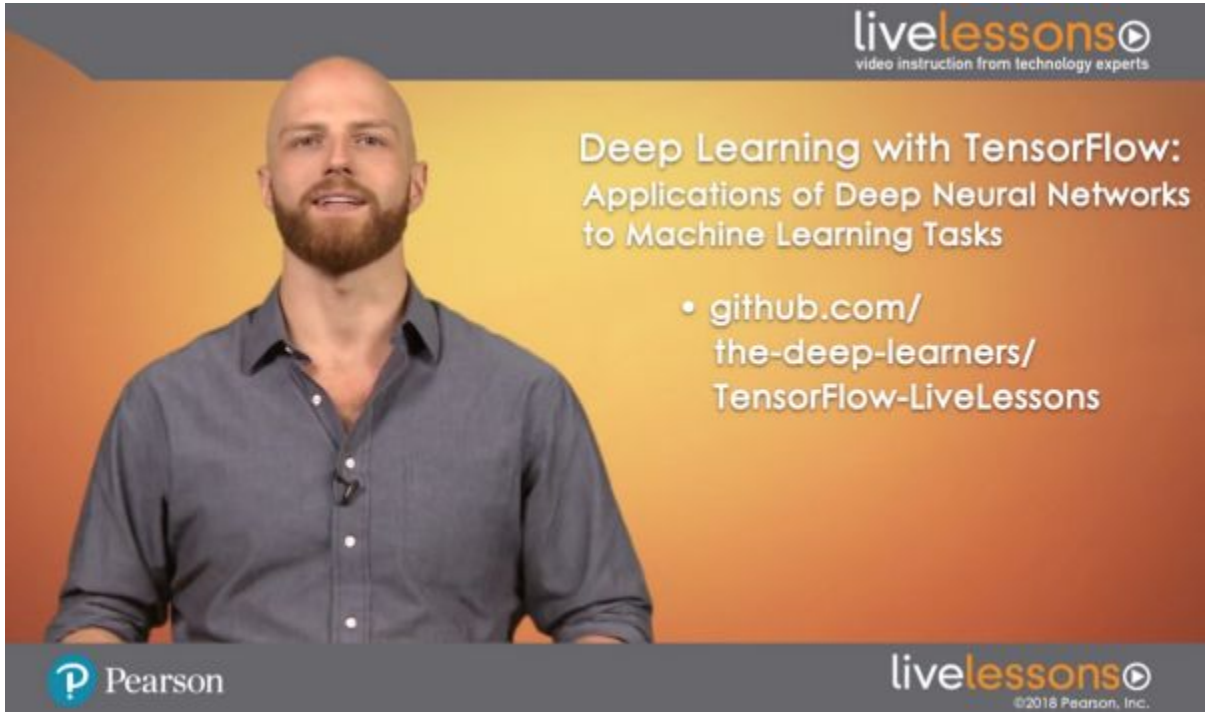
- Running them on Unix/Linux
- Running them on Mac
- Running them on Windows
- Viewing only, e.g., on GitHub
- Other

Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. Deep Learning with Keras
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

Deep Learning Fundamentals

1. **The Unreasonable Effectiveness of Deep Learning**
   - Intro to Neural Networks and Deep Learning
   - The Deep Learning Families and Libraries
   - Learning with Artificial Neurons
   - TensorFlow Playground
2. Deep Learning with Keras
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

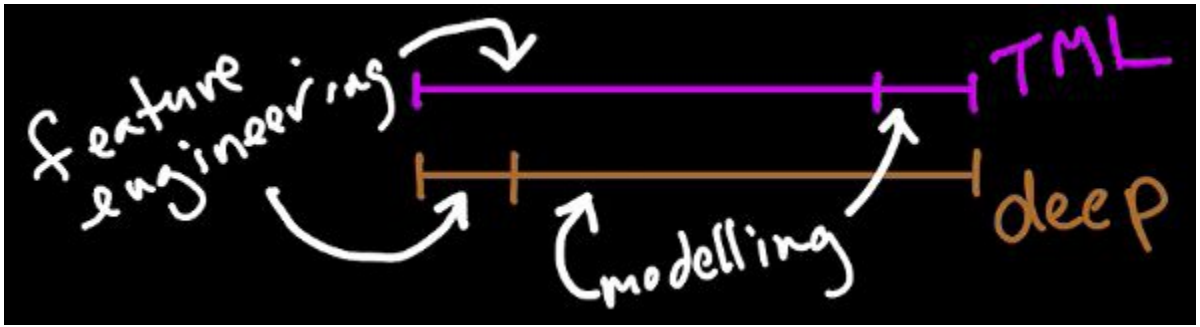


Search *Deep Learning with TensorFlow* or *Jon Krohn* in Safari

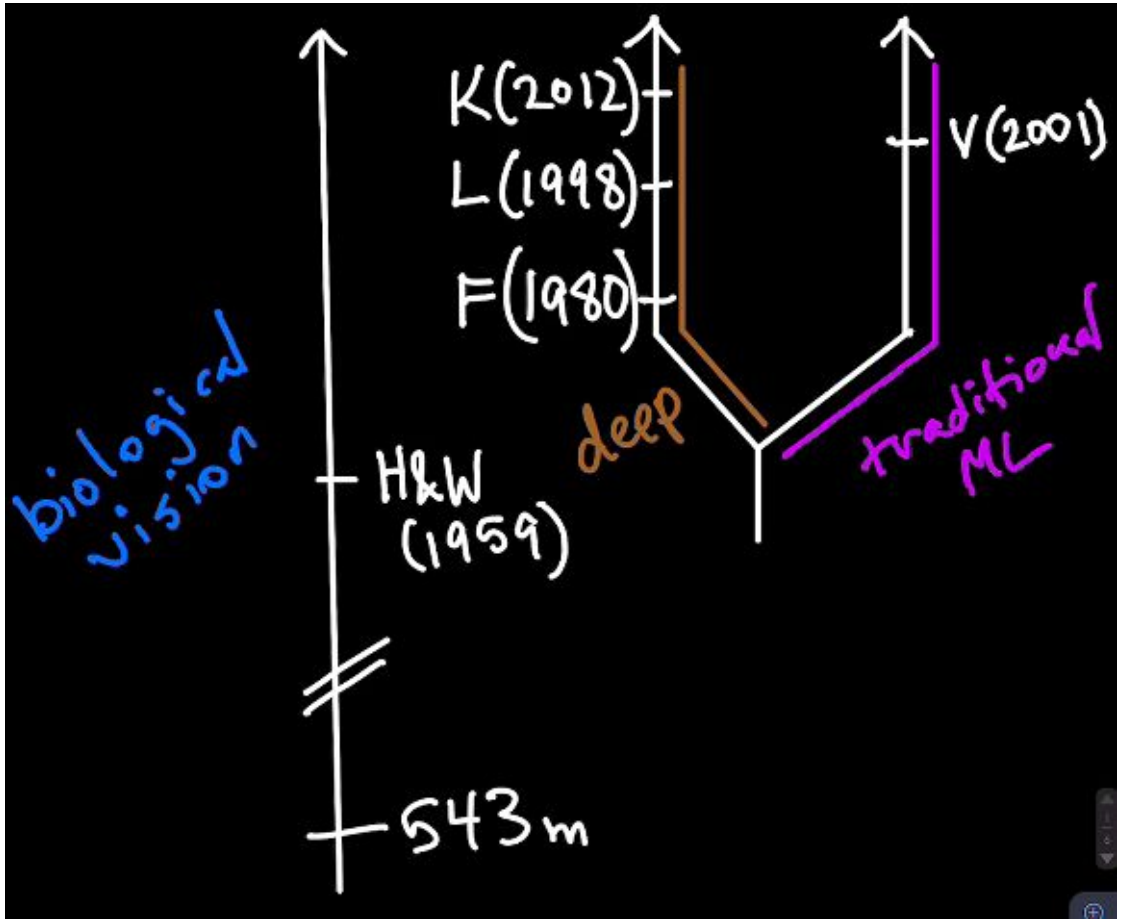
Deep Learning Fundamentals

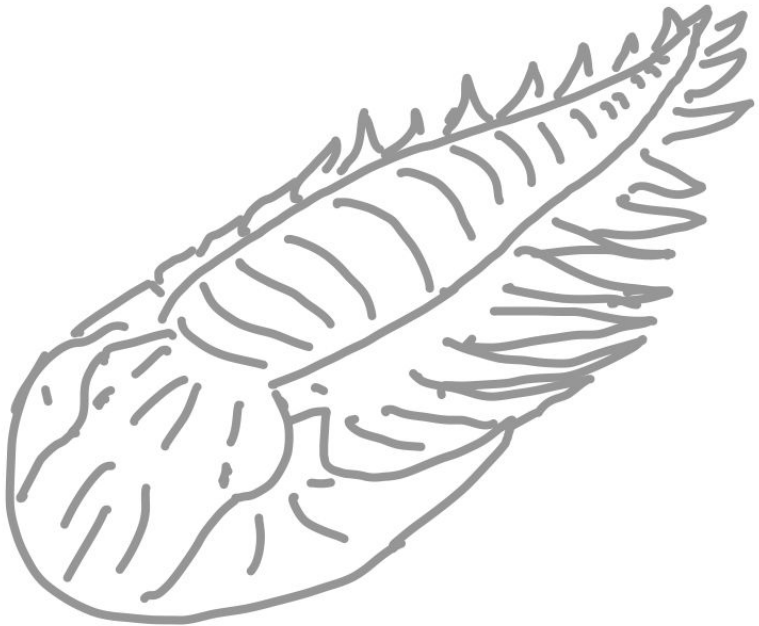
1. **The Unreasonable Effectiveness of Deep Learning**
   - **Intro to Neural Networks and Deep Learning** *(reference LiveLessons section 1.1)*
   - The Deep Learning Families and Libraries
   - Learning with Artificial Neurons
   - TensorFlow Playground
2. Deep Learning with Keras
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

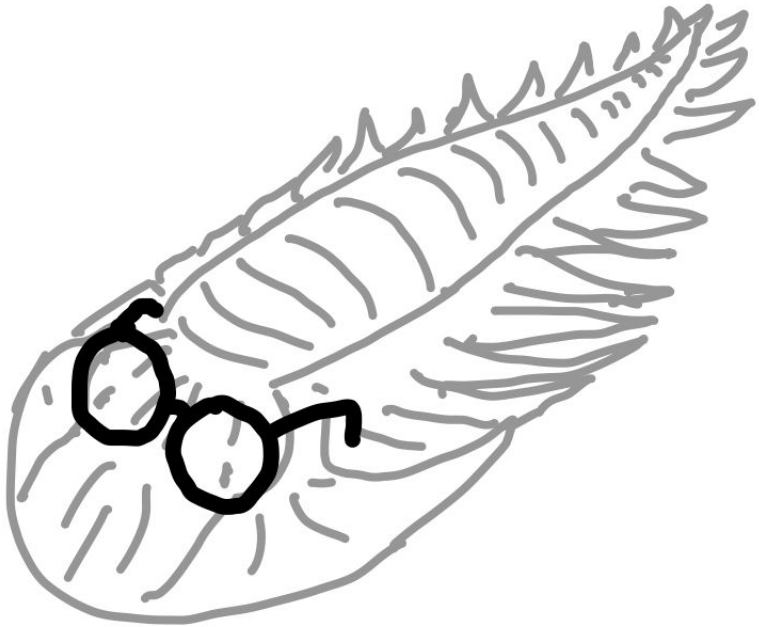
Traditional ML vs Deep Learning

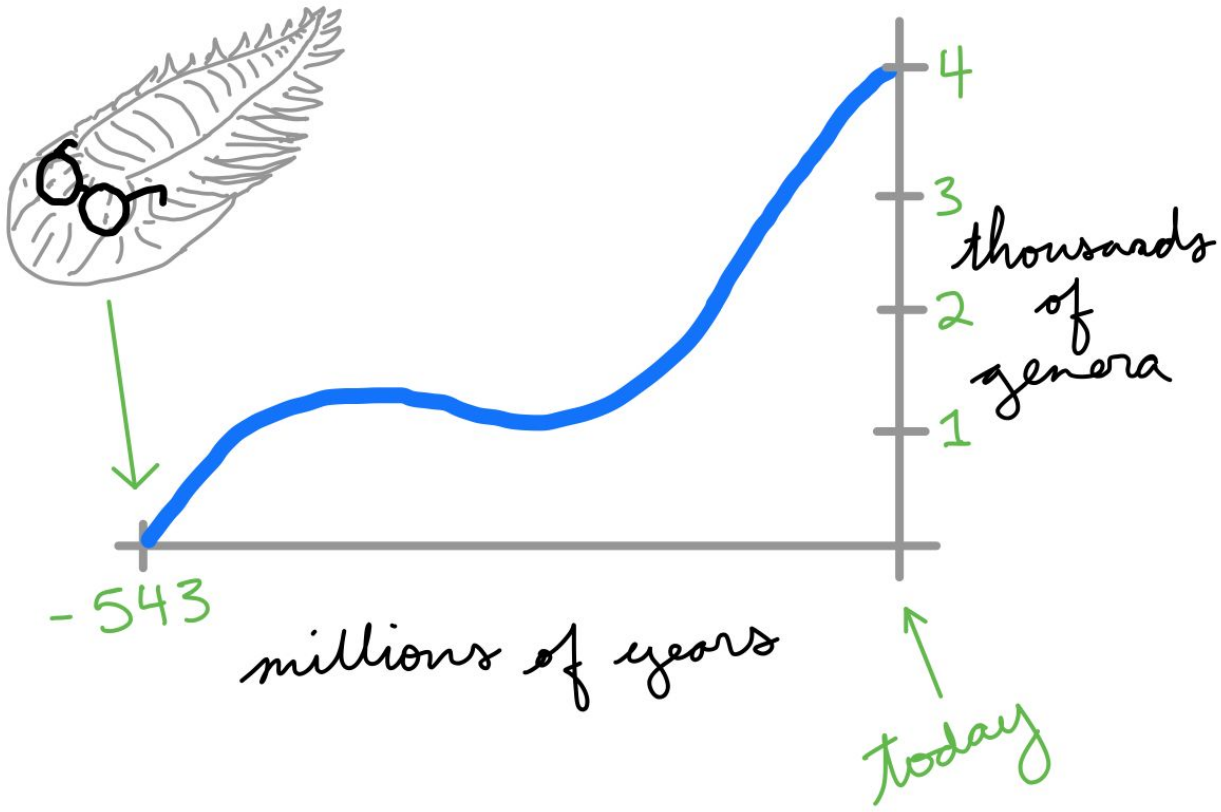


Case Study: The History of Vision

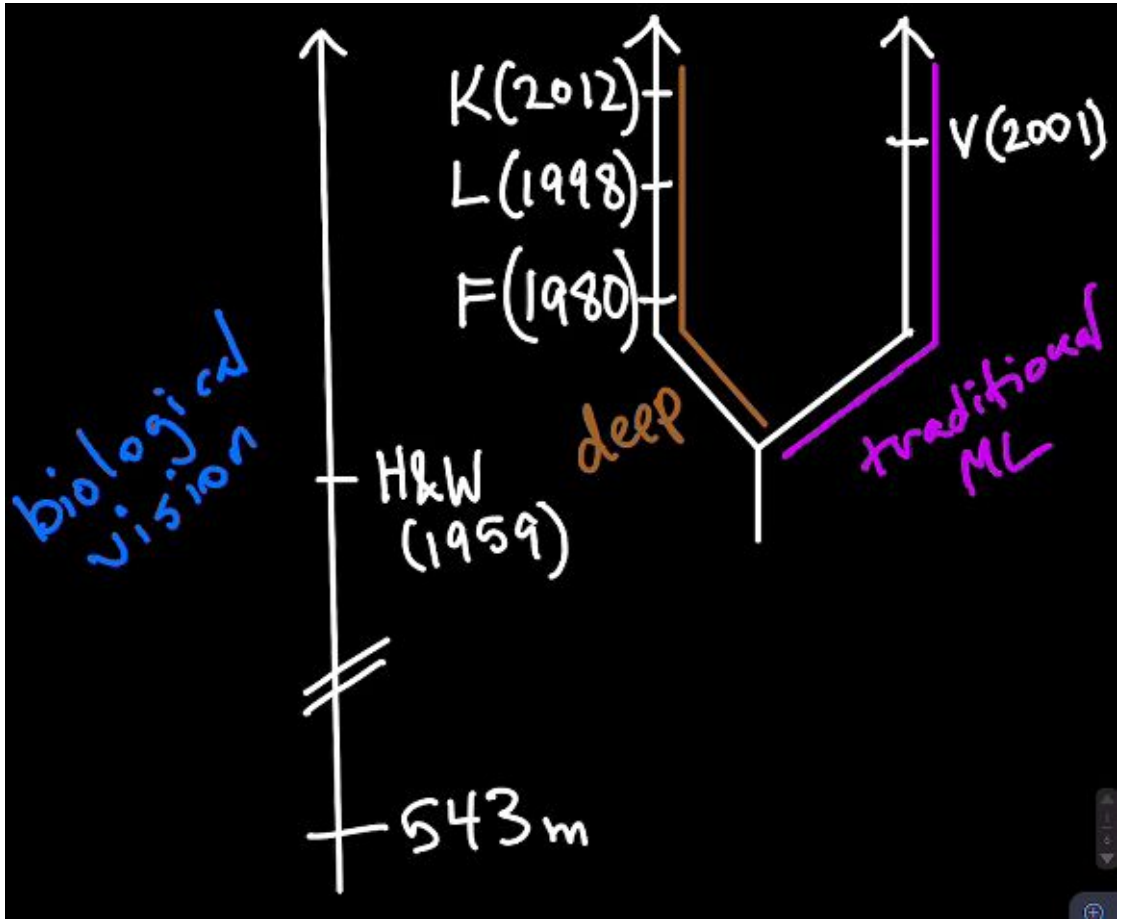


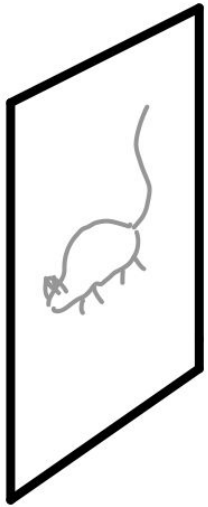


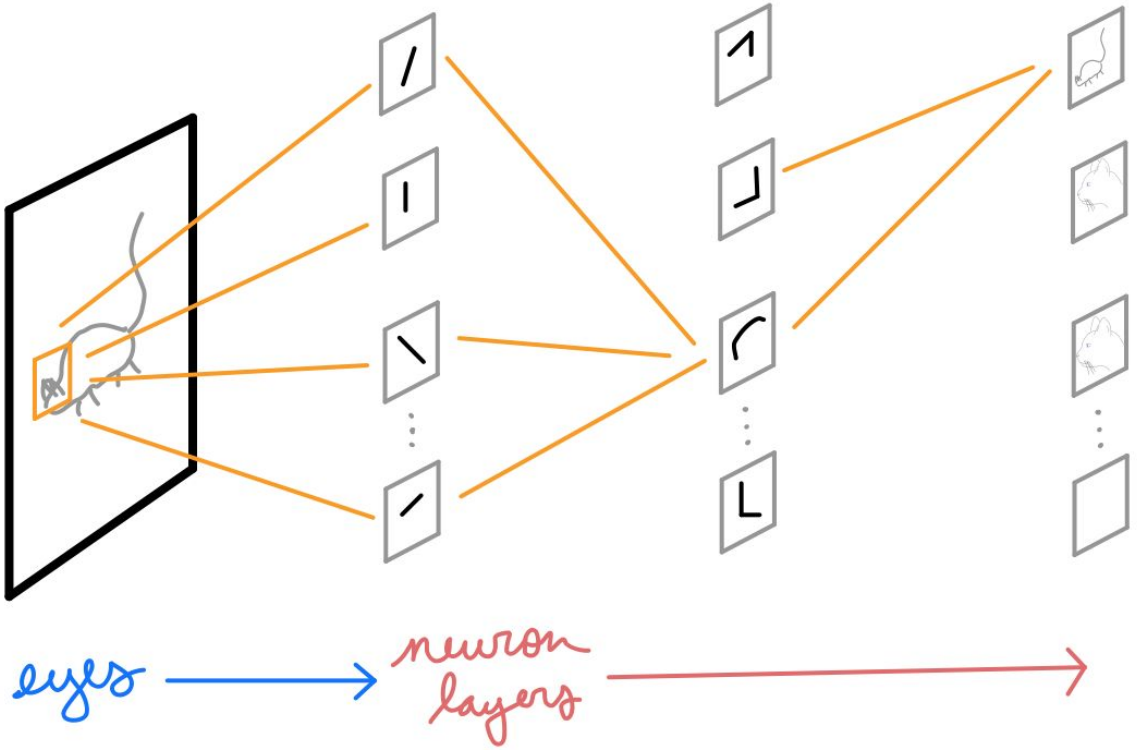




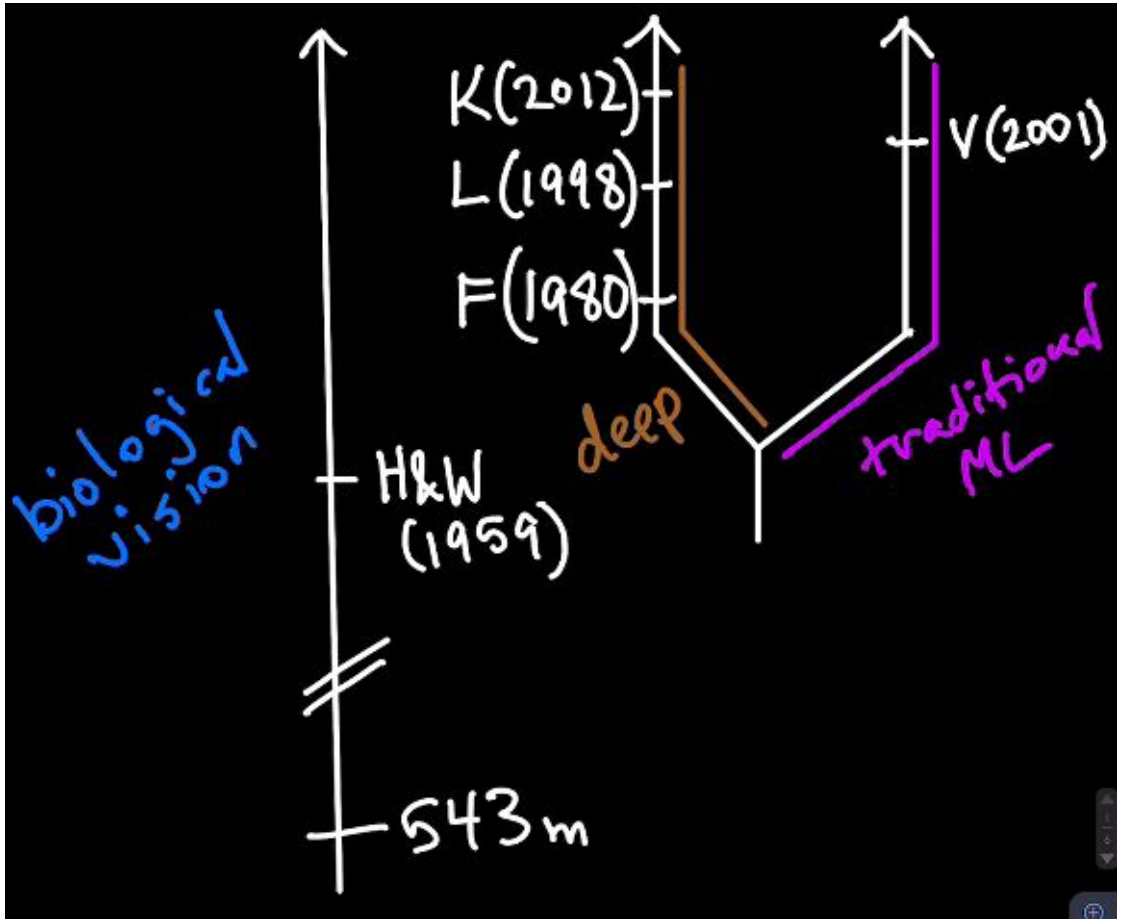
Case Study: The History of Vision







Case Study: The History of Vision



Neurocognitron (Fukushima, 1980)

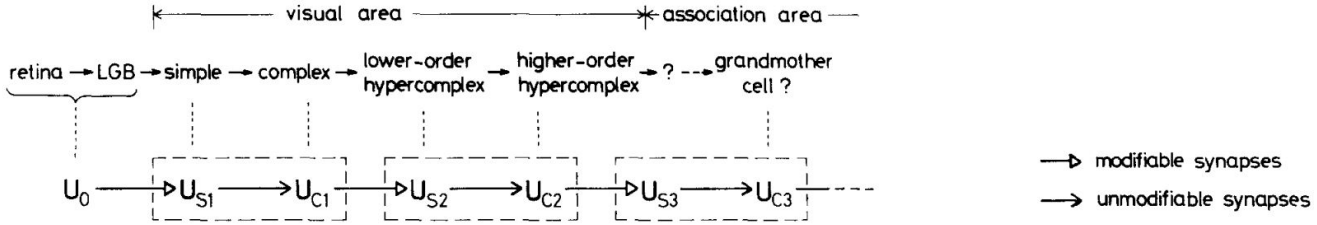


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

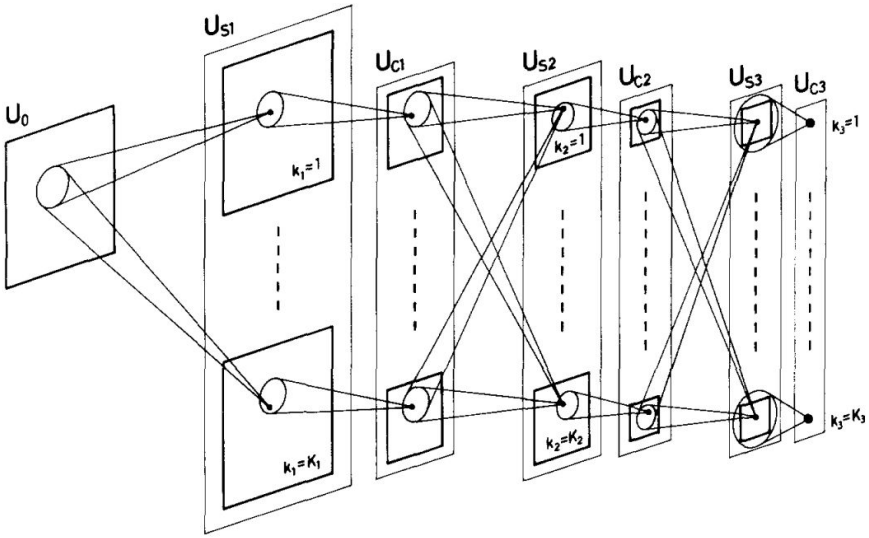
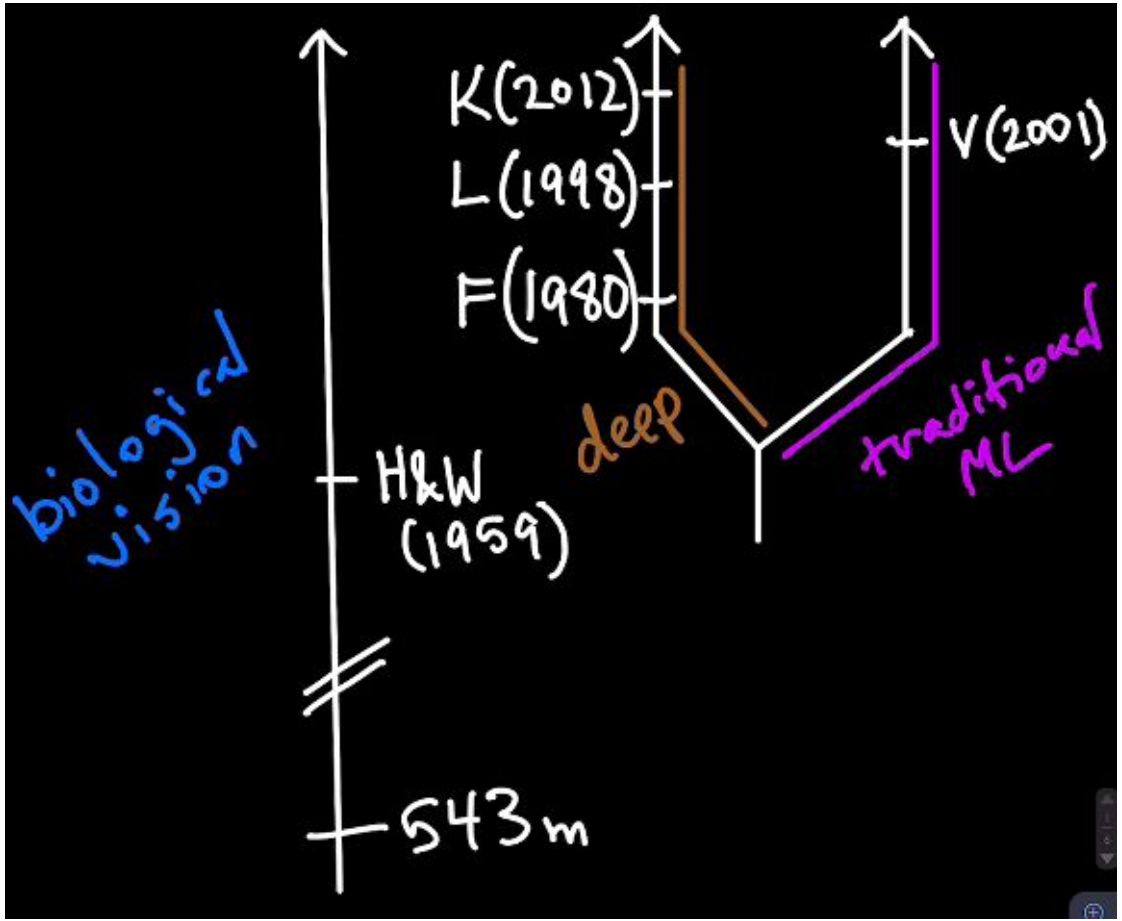
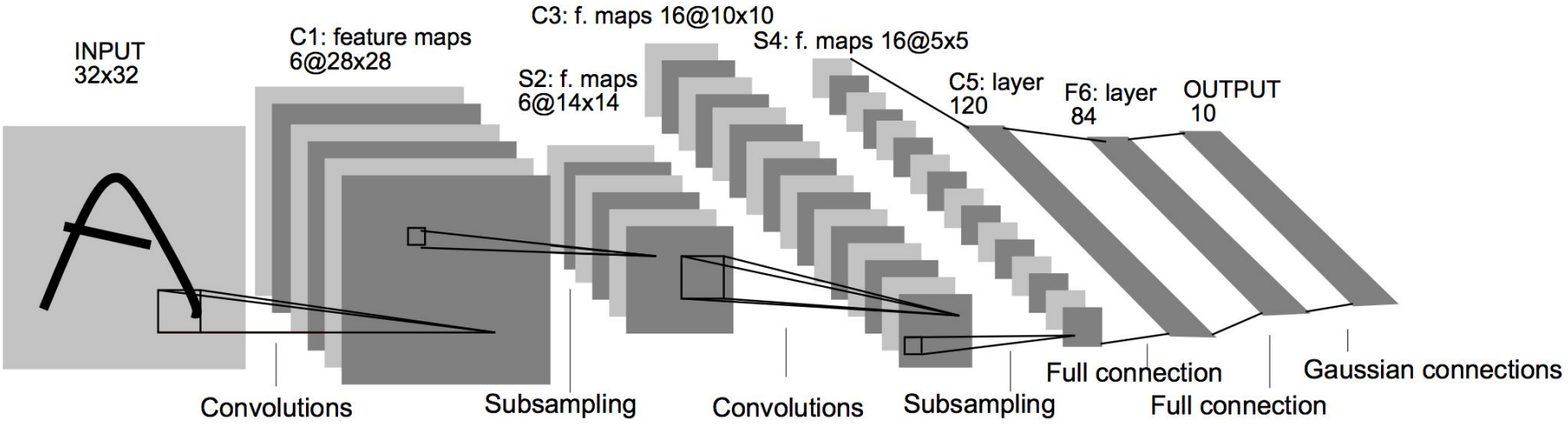
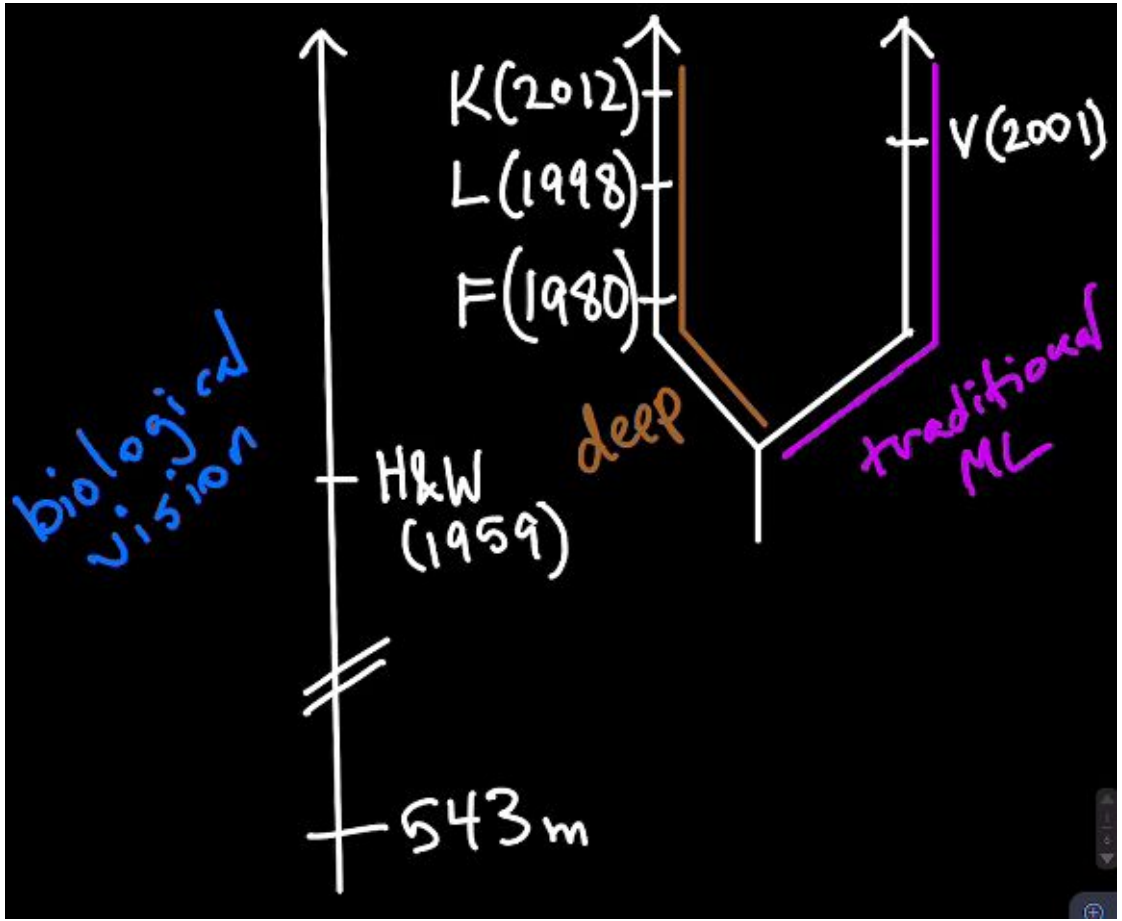


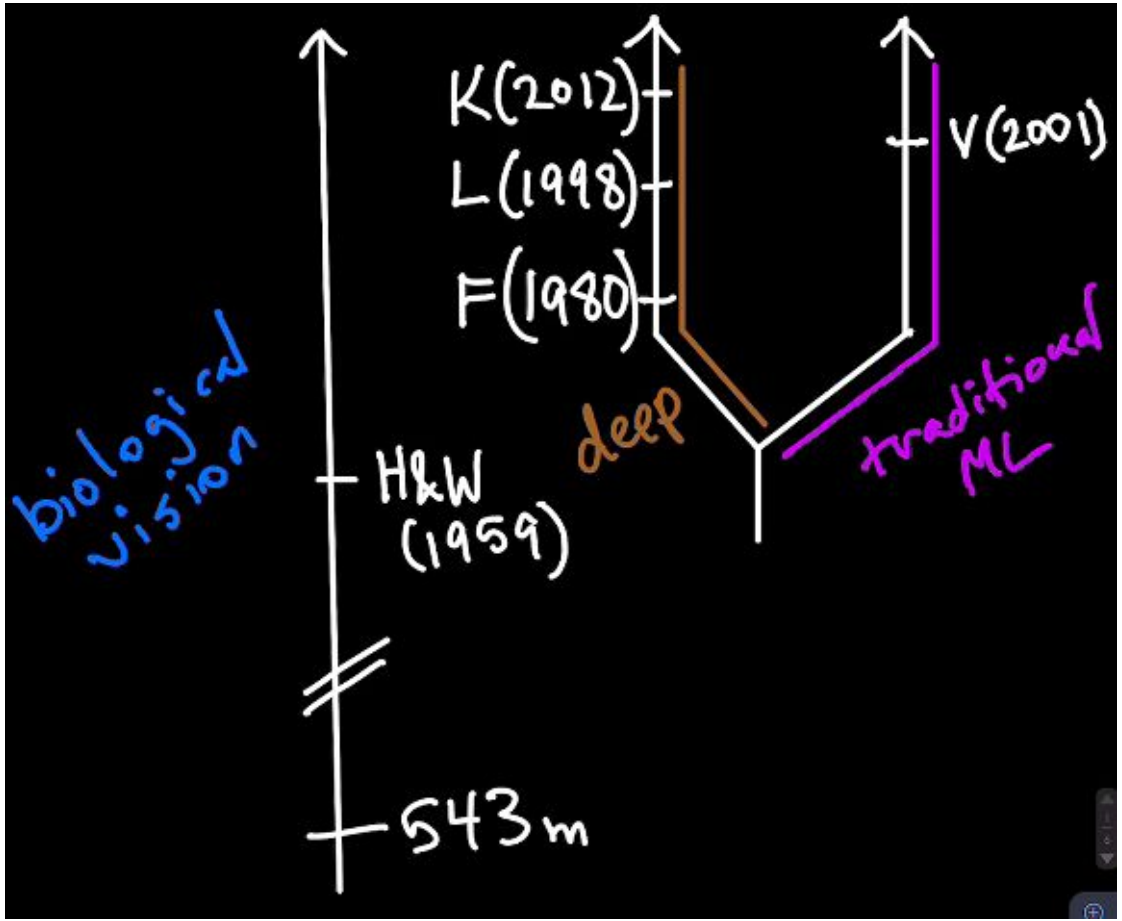
Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron



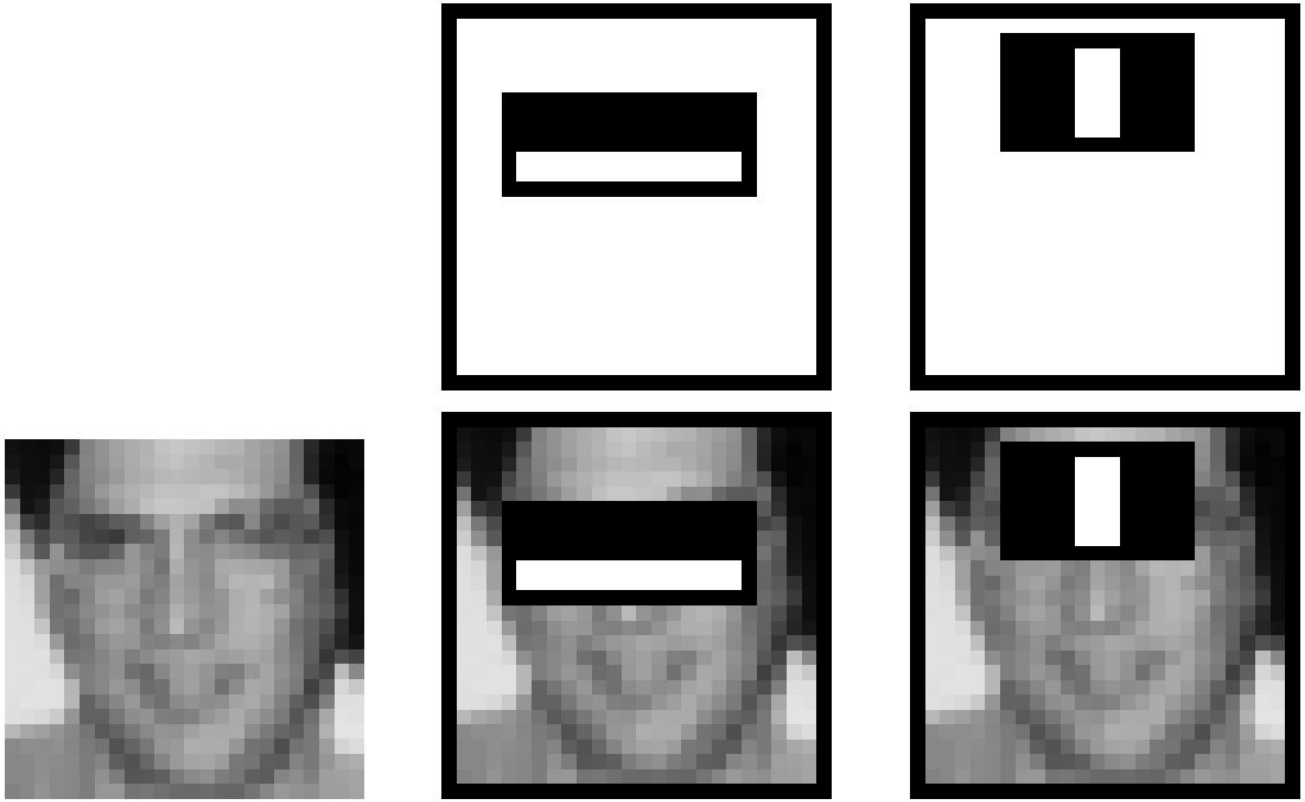
LeNet-5 (LeCun et al., 1998)

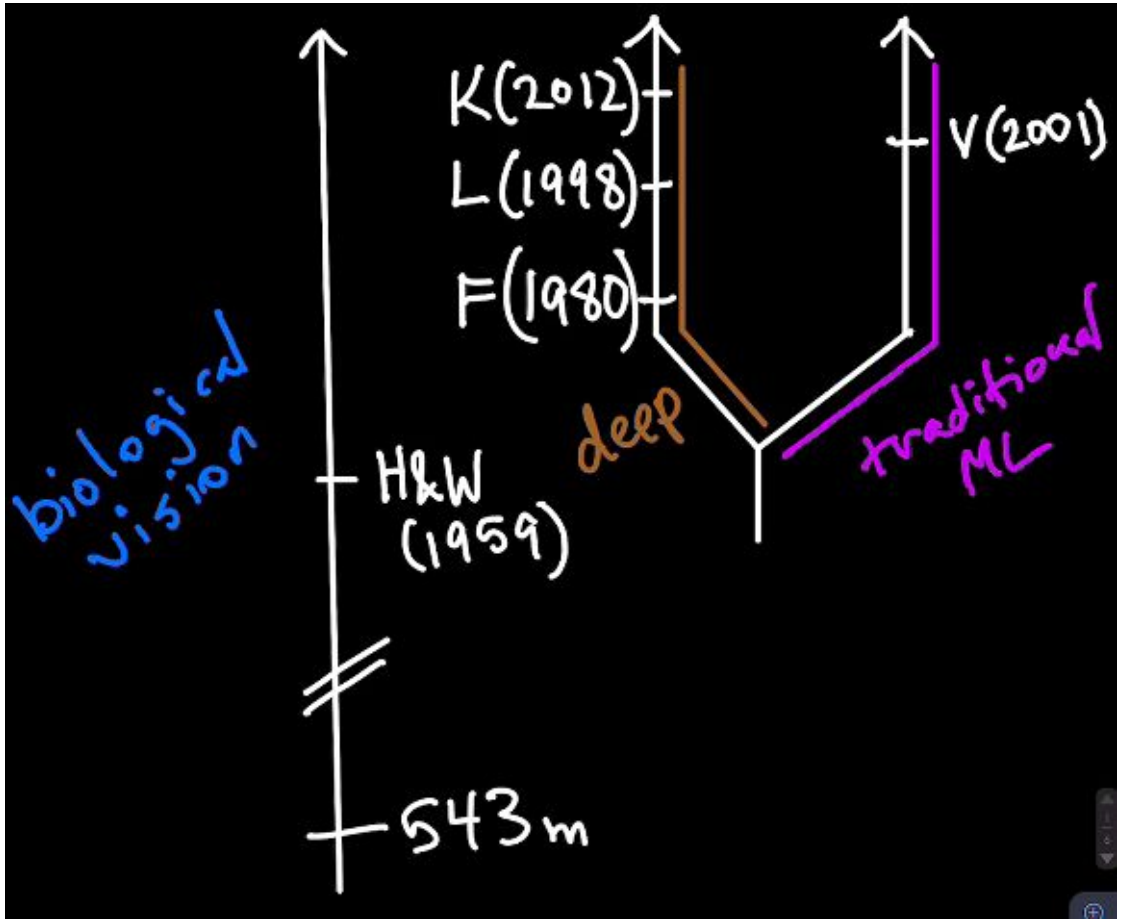




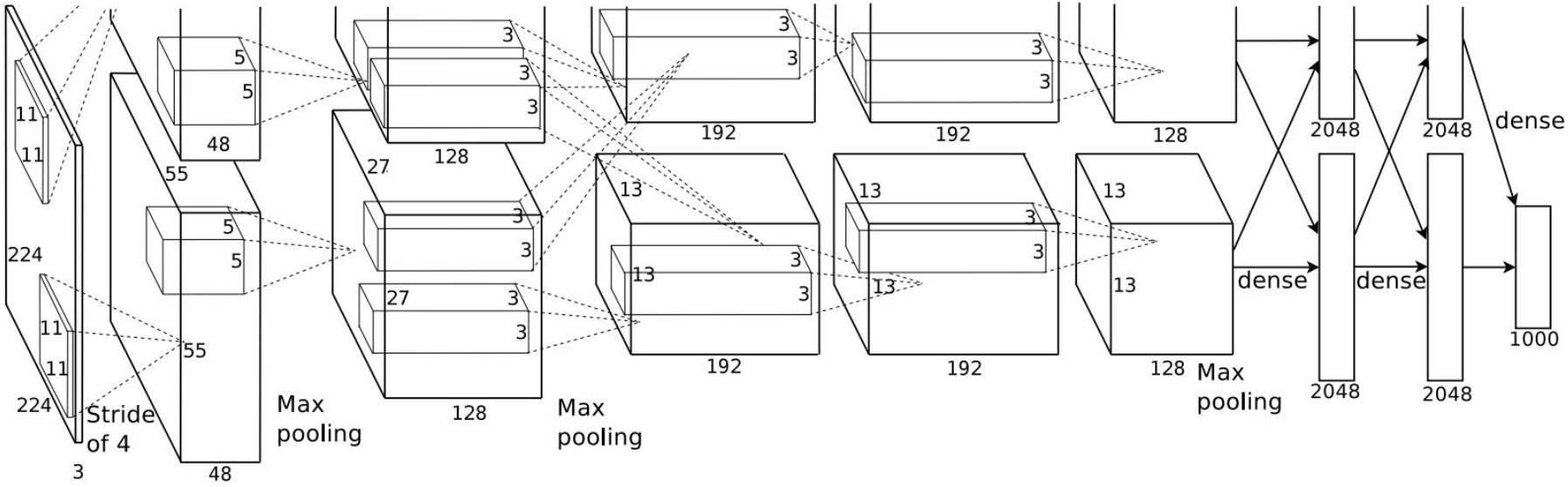


Viola & Jones (2001)





AlexNet (Krizhevsky et al., 2012)



POLL

If a voice recognition algorithm is fed audio of speech as inputs, given corresponding text as the outputs (labels) to learn, and no features are explicitly programmed, is this a:

- Traditional Machine Learning Algorithm
- Deep Learning Algorithm
- I Don't Know

Deep Learning Fundamentals

1. **The Unreasonable Effectiveness of Deep Learning**
   - Intro to Neural Networks and Deep Learning
   - **The Deep Learning Families and Libraries** *(reference LiveLessons sections 2.1 & 4.1)*
   - Learning with Artificial Neurons
   - TensorFlow Playground
2. Deep Learning with Keras
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

Dense Networks

ConvNets: Convolutional Networks

RNNs: Recurrent Neural Networks

Deep Reinforcement Learning

GANs: Generative Adversarial Networks

POLL

If you were designing an algorithm to learn to play Tetris by maximizing its score, which of these Deep Learning approaches would be most appropriate?

- Convolutional Neural Network
- Recurrent Neural Network
- Deep Reinforcement Learning
- Generative Adversarial Network

POLL

If you were designing an algorithm to recognise tumours in medical images, which of these Deep Learning approaches would be most appropriate?

- Convolutional Neural Network
- Recurrent Neural Network
- Deep Reinforcement Learning
- Generative Adversarial Network

POLL

If you were designing an algorithm to predict stock price movements based on time series data, which of these Deep Learning approaches would be most appropriate?

- Convolutional Neural Network
- Recurrent Neural Network
- Deep Reinforcement Learning
- Generative Adversarial Network

Leading Deep Learning Libraries

	Caffe	Torch	MXNet	TensorFlow
Language	Python, Matlab	Lua, C	Python, R, C++ Julia, Matlab JavaScript, Go Scala, Perl	Python, R, C++ C, Java, Go
Programming Style	Symbolic	Imperative	Imperative	Symbolic
Parallel GPUs: Data	Yes	Yes	Yes	Yes
Parallel GPUs: Model		Yes	Yes	Yes
Pre-Trained Models	Model Zoo	ModelZoo	Model Zoo	github.com/tensorflow/models
For RNNs				Best
High-Level APIs		PyTorch	in-built	Keras, TFLearn

Deep Learning Fundamentals

1. **The Unreasonable Effectiveness of Deep Learning**
   - Intro to Neural Networks and Deep Learning
   - The Deep Learning Families and Libraries
   - **Learning with Artificial Neurons** *(reference LiveLessons sections 2.2, 2.3 & 3.1)*
   - **TensorFlow Playground** *(2.4)*
2. Deep Learning with Keras
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

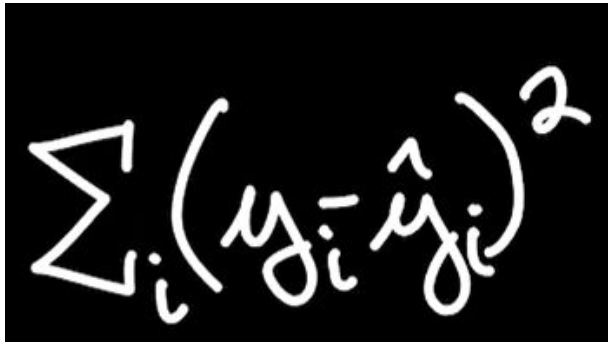
Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

Cost Functions

- quadratic cost



Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

Cost Functions

- quadratic cost
- cross-entropy

Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

Cost Functions

- quadratic cost
- cross-entropy

Gradient Descent

Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

Cost Functions

- quadratic cost
- cross-entropy

Gradient Descent

Backpropagation

Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

Cost Functions

- quadratic cost
- cross-entropy

Gradient Descent

Backpropagation

Layers

- dense
- softmax

Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

Cost Functions

- quadratic cost
- cross-entropy

Gradient Descent

Backpropagation

Layers

- dense
- softmax

Initialization

- Glorot normal
- Glorot uniform

Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

Cost Functions

- quadratic cost
- cross-entropy

***Stochastic* Gradient Descent**

- mini-batch size
- learning rate
- second-order, e.g., Adam

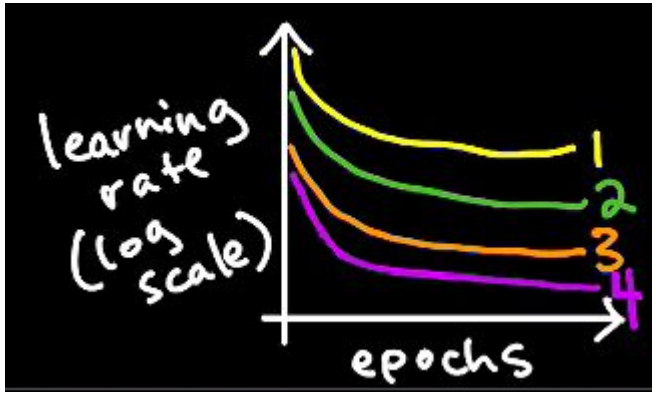
Backpropagation

Initialization

- Glorot normal
- Glorot uniform

Layers

- dense
- softmax



Your Arsenal

Neurons

- sigmoid
- tanh
- ReLU

Cost Functions

- quadratic cost
- cross-entropy

***Stochastic* Gradient Descent**

- mini-batch size
- learning rate
- second-order, e.g., Adam

Backpropagation

Initialization

- Glorot normal
- Glorot uniform

Layers

- dense
- softmax

Avoiding Overfitting

- L1/L2 Regularization
- Dropout
- Data Expansion

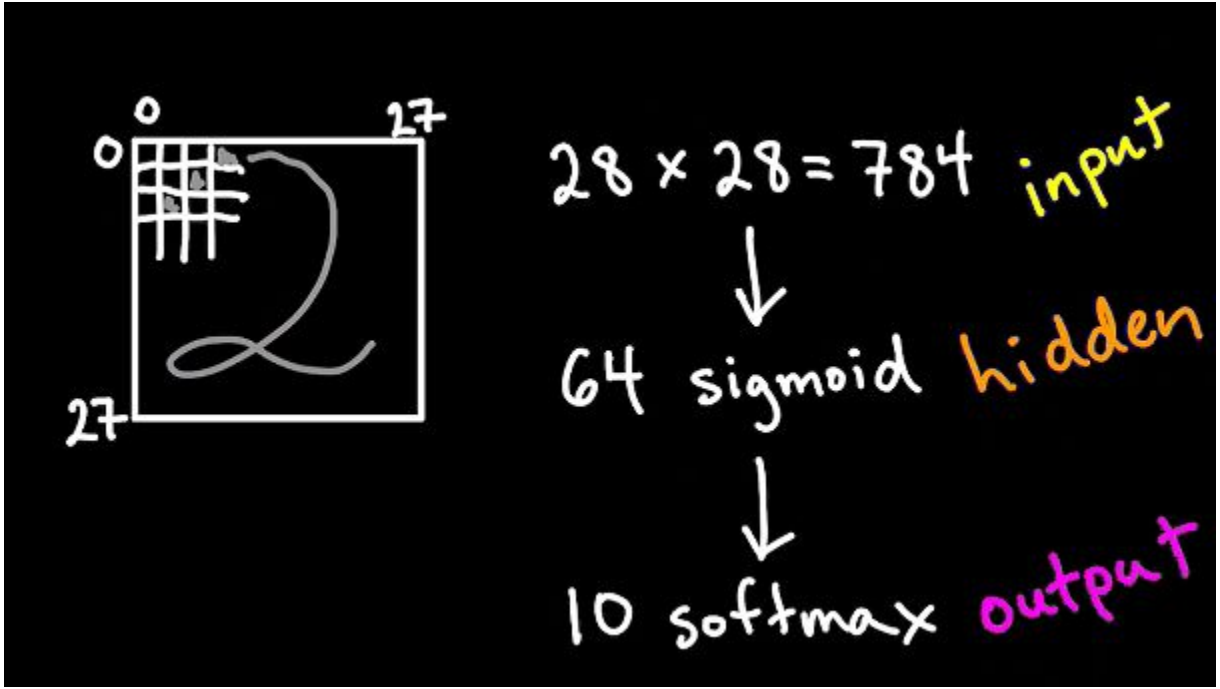
Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. **Deep Learning with Keras**
   - **Shallow Neural Networks**
   - **Deep Neural Networks**
   - **How to Build Your Own Project**
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. **Deep Learning with Keras**
   - **Shallow Neural Networks** *(LiveLessons 1.3 & 2.5)*
   - Deep Neural Networks
   - How to Build Your Own Project
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

MNIST Digits



Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. **Deep Learning with Keras**
   - Shallow Neural Networks
   - Deep Neural Networks *(LiveLessons section 3.2)*
   - How to Build Your Own Project
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. **Deep Learning with Keras**
   - Shallow Neural Networks
   - Deep Neural Networks
   - How to Build Your Own Project *(LiveLessons 5.1 & 5.2)*
3. Introduction to TensorFlow
4. Deep Learning with TensorFlow

How to Build Your Own DL Project

1. Define Task
2. Define Data Set
3. Define Metric
4. Split Data
5. Establish Baseline
6. Implement Existing
7. Improve...

Tuning Hyperparameters

1. Get Above Chance
2. Experiment with Layers (Count, Variety)
3. Glorot Initialization
4. Choose Cost Function
5. Avoid Overfitting
6. Vary Learning Rate
7. Epochs
8. Regularization λ
9. Batch Size
10. Automation

Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. Deep Learning with Keras
3. **Introduction to TensorFlow**
   - **TensorFlow Graphs**
   - **Neurons in TensorFlow**
4. Deep Learning with TensorFlow

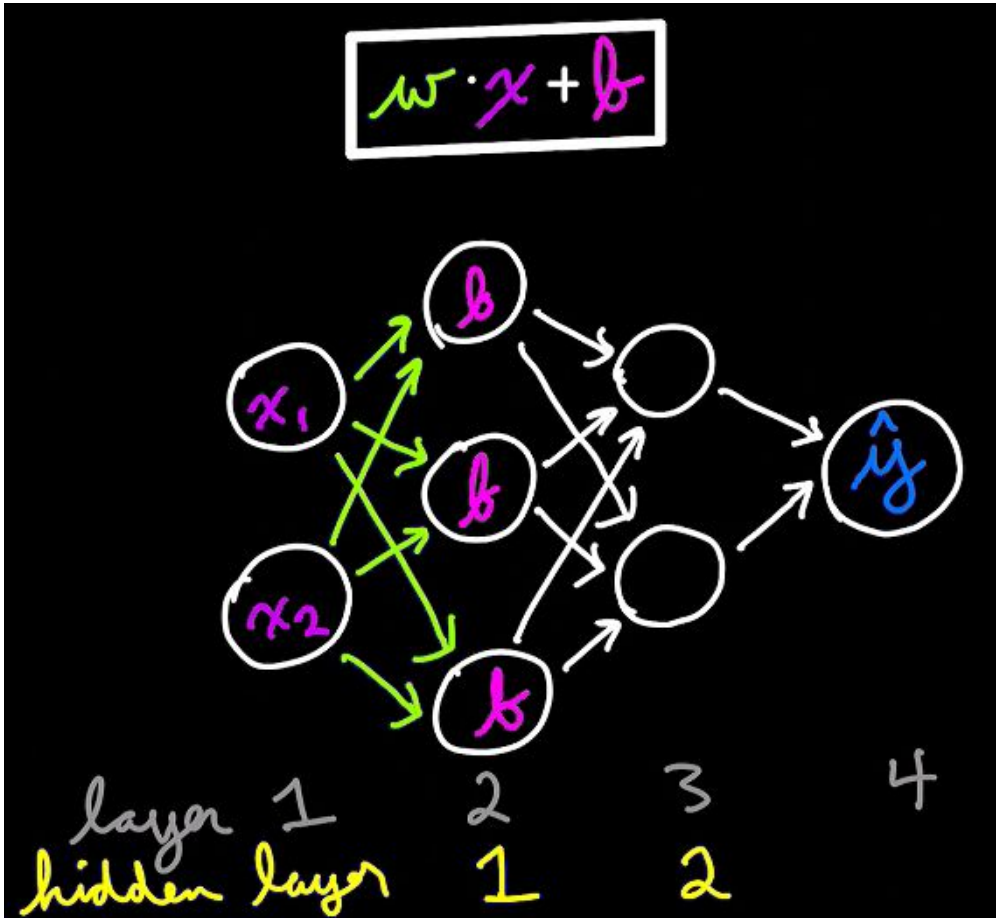
Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. Deep Learning with Keras
3. **Introduction to TensorFlow**
   - **TensorFlow Graphs** *(LiveLessons section 4.2)*
   - Neurons in TensorFlow
4. Deep Learning with TensorFlow

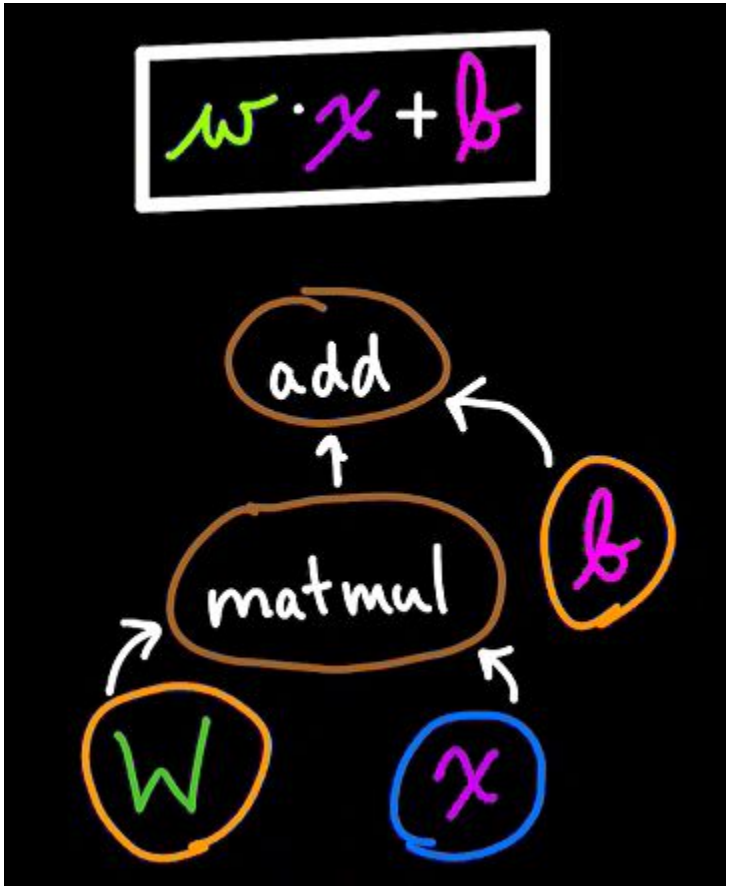
TensorFlow Graphs

1. build graph
2. initialize session
3. fetch and feed data

A Familiar Equation



TensorFlow Graphs



POLL

in a TensorFlow graph, the calculation of a mean is best associated with:

- an operation
- a Variable tensor
- a placeholder tensor

POLL

in a TensorFlow graph, the labels we're predicting with our model are best associated with:

- an operation
- a Variable tensor
- a placeholder tensor

POLL

in a TensorFlow graph, the slope of a regression line is best associated with:

- an operation
- a Variable tensor
- a placeholder tensor

Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. Deep Learning with Keras
3. **Introduction to TensorFlow**
   - TensorFlow Graphs
   - **Neurons in TensorFlow** *(LiveLessons section 4.2)*
4. Deep Learning with TensorFlow

Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. Deep Learning with Keras
3. Introduction to TensorFlow
4. **Deep Learning with TensorFlow**
   - Model-Fitting
   - Deep Neural Nets

Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. Deep Learning with Keras
3. Introduction to TensorFlow
4. **Deep Learning with TensorFlow**
   - **Model-Fitting** *(LiveLessons 4.3)*
   - Deep Neural Nets

Deep Learning Fundamentals

1. The Unreasonable Effectiveness of Deep Learning
2. Deep Learning with Keras
3. Introduction to TensorFlow
4. **Deep Learning with TensorFlow**
   - Model-Fitting
   - **Deep Neural Nets** *(LiveLessons 4.4)*

Sneak Peak at *LiveLessons 3.5*

Design neural network architecture

In [4]:
```
model = Sequential()

model.add(Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activation='relu', input_shape=(224, 22
4, 3)))
model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))
model.add(BatchNormalization())

model.add(Conv2D(256, kernel_size=(5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))
model.add(BatchNormalization())

model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(384, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(384, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))
model.add(BatchNormalization())

model.add(Flatten())
model.add(Dense(4096, activation='tanh'))
model.add(Dropout(0.5))
model.add(Dense(4096, activation='tanh'))
model.add(Dropout(0.5))

model.add(Dense(17, activation='softmax'))
```

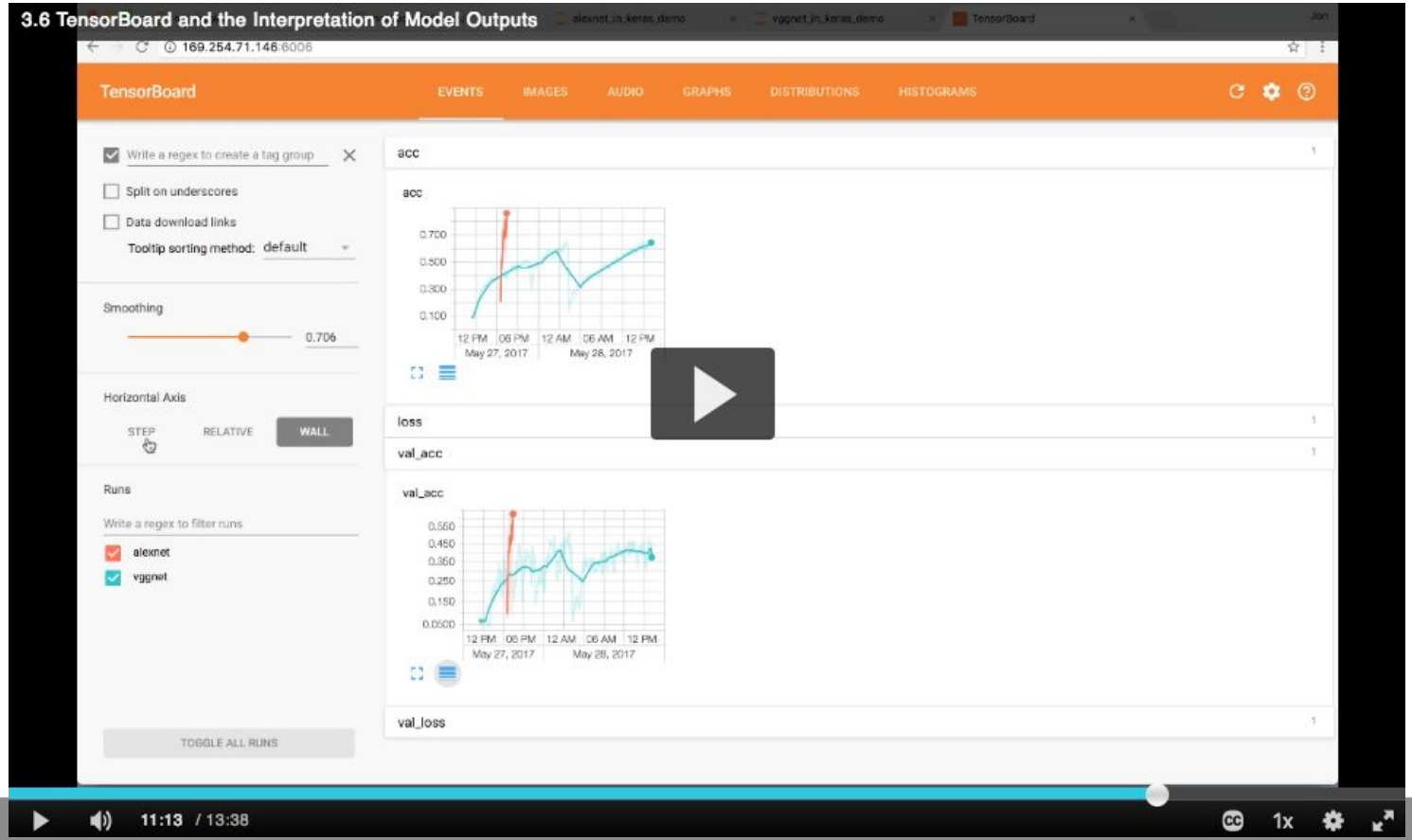
Sneak Peak at *LiveLessons 4.5*

Design neural network architecture

In [7]:

```
def network(x, weights, biases, n_in, mp_psize, mp_dropout, dense_dropout):

    # reshape linear MNIST pixel input into square image:
    square_dimensions = int(np.sqrt(n_in))
    square_x = tf.reshape(x, shape=[-1, square_dimensions, square_dimensions, 1])

    # convolutional and max-pooling layers:
    conv_1 = conv2d(square_x, weights['W_c1'], biases['b_c1'])
    conv_2 = conv2d(conv_1, weights['W_c2'], biases['b_c2'])
    pool_1 = maxpooling2d(conv_2, mp_psize)
    pool_1 = tf.nn.dropout(pool_1, 1-mp_dropout)

    # dense layer:
    flat = tf.reshape(pool_1, [-1, weights['W_d1'].get_shape().as_list()[0]])
    dense_1 = dense(flat, weights['W_d1'], biases['b_d1'])
    dense_1 = tf.nn.dropout(dense_1, 1-dense_dropout)

    # output layer:
    out_layer_z = tf.add(tf.matmul(dense_1, weights['W_out']), biases['b_out'])

    return out_layer_z
```

Sneak Peak at *LiveLessons 3.6*



POLL

what's most useful for follow-up?

- CNNs and Machine Vision
- Sequences: RNNs, LSTMs, NLP, Financial Time Series
- Generative Adversarial Networks
- Reinforcement Learning
- TensorBoard
- more detail on fundamental theory
- Something Else

Search *Deep Learning for Natural Language Processing* or *Jon Krohn* in Safari