Open Source Models
The Next Step Towards Easy AI

Yasen Kiprov

yasen.kiprov@gmail.com

About Me

- AI Team Leader at SiteGround
- Lots of experience in Natural Language Processing
- Lots of experience as a developer
- R&D with focus on AI applications
- Co-organizer of PyData Sofia Meetups

I don’t like modern NN architectures - they are too complex

About PyData Sofia

- **NumFOCUS**: Home of NumPy, SciPy, Matplotlib, Pandas, etc.
- **PyData**: educational program of NumFOCUS
- Monthly meetups. Next is on **5.12.2019**
- Speakers wanted
  - beginner to expert level
  - best practices, new approaches, emerging technologies
  - ML, data management, processing, analytics, and visualization
- Hosts wanted
- Find us on Meetup and Facebook

What is a Neuron

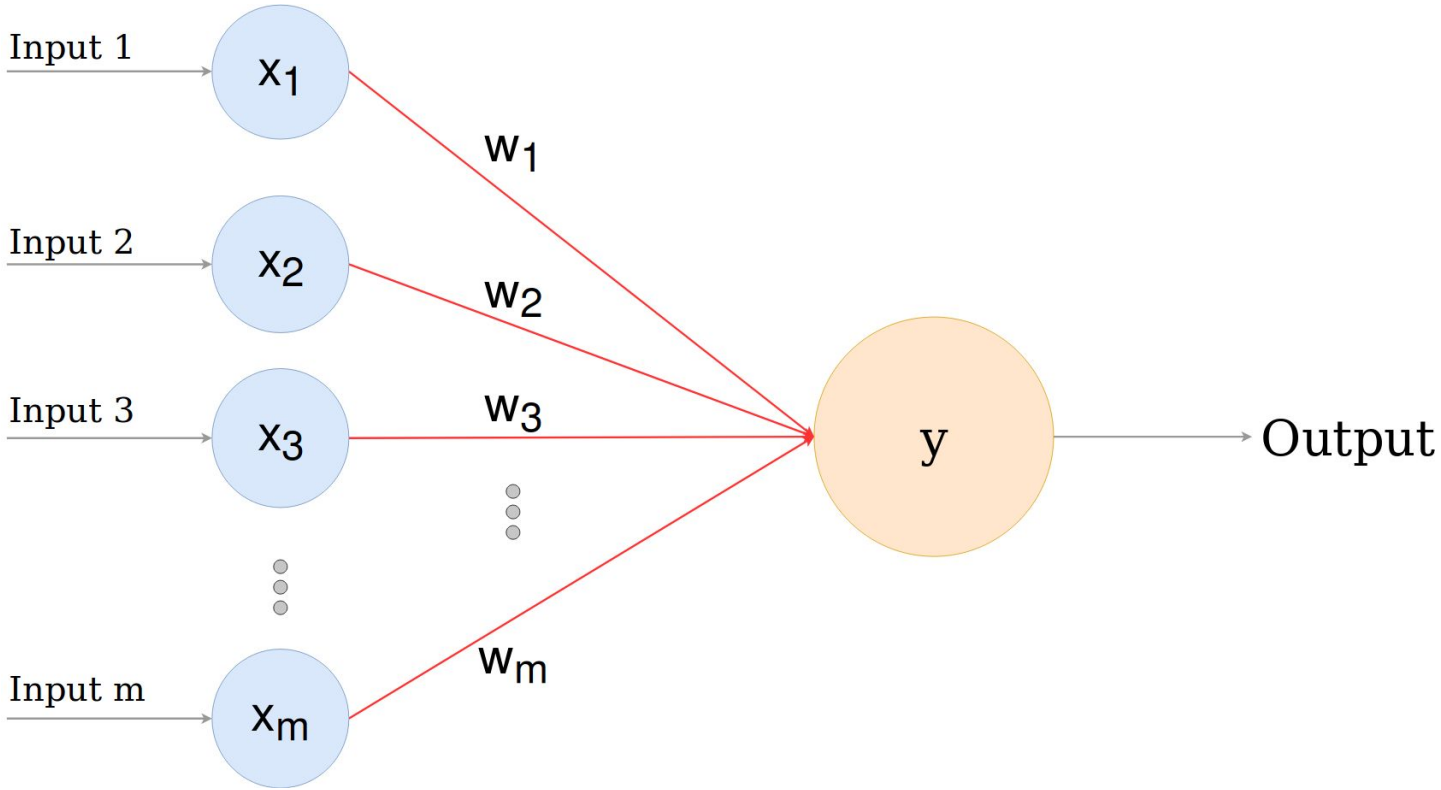


Image source: https://towardsdatascience.com/the-perceptron-3af34c84838c

What is a Neural Network

Fully connected

Useful for classification

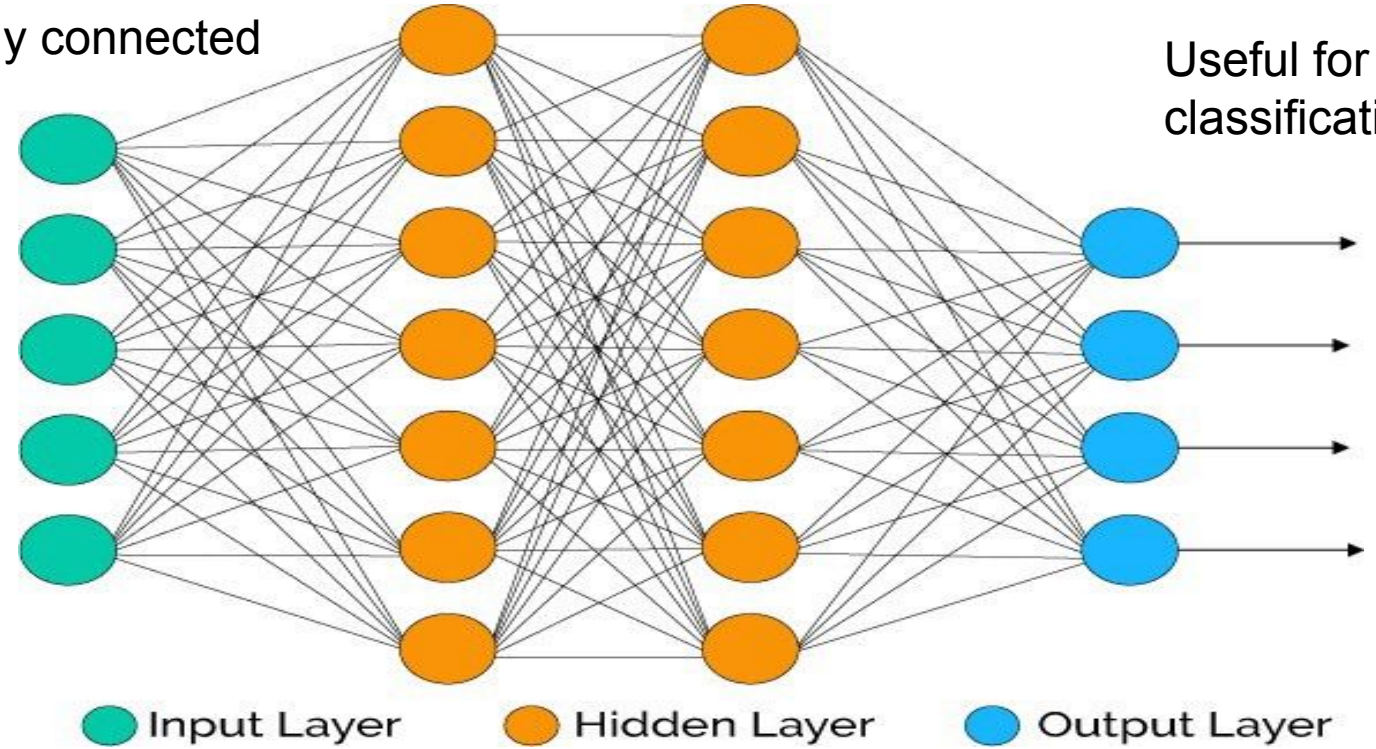


Image source: https://medium.com/datadriveninvestor/when-not-to-use-neural-networks-89fb50622429

Weights

Input to hidden layer weights

W11	W12	W13	W14
W21	W22	W23	W24
W31	W32	W33	W34

Hidden to output layer weights

W71	W72	W73
W81	W82	W83
W91	W92	W93

Hidden layer biases

B1
B2
B3

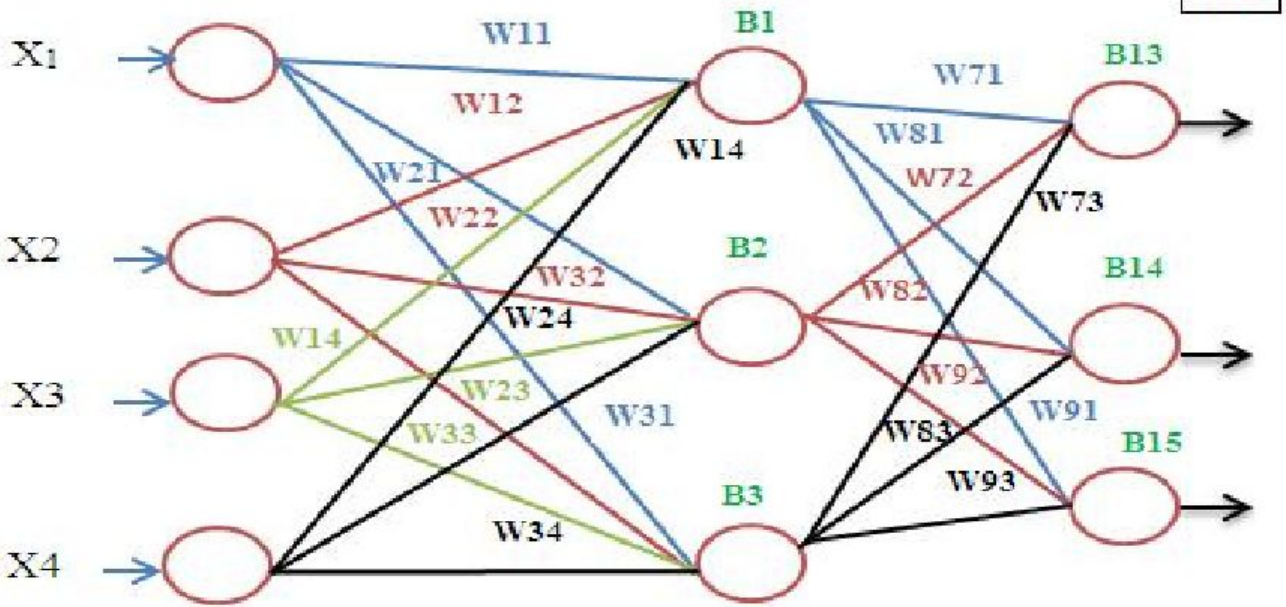
Output layer biases

B13
B14
B15

Forward pass (predict)

Backpropagation (update W)

Generalize (unseen data)



Convolutional NN For Image Classification

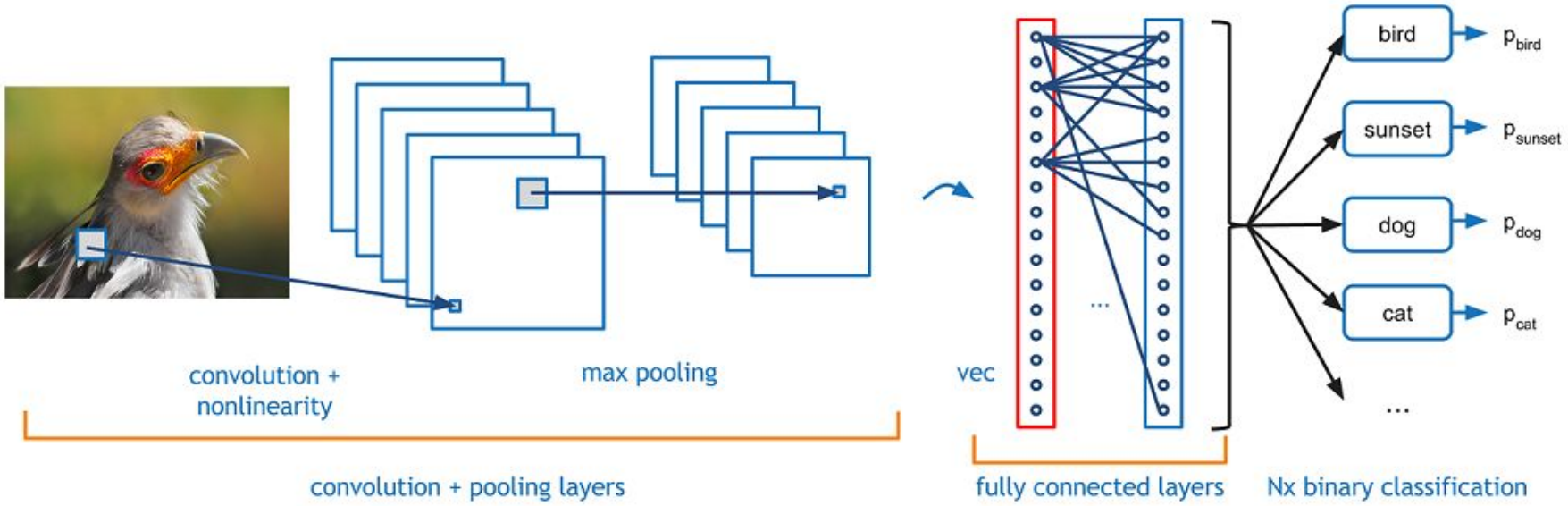


Image source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Layers

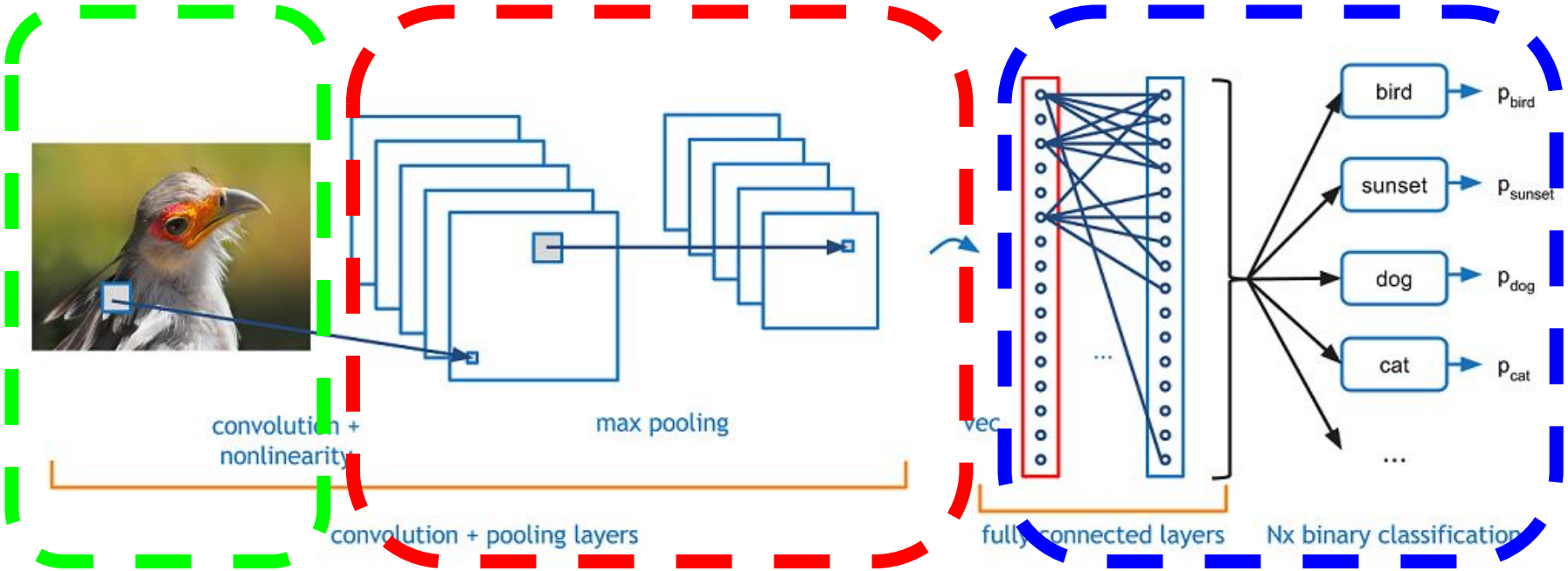


Image source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Text Classification with a CNN

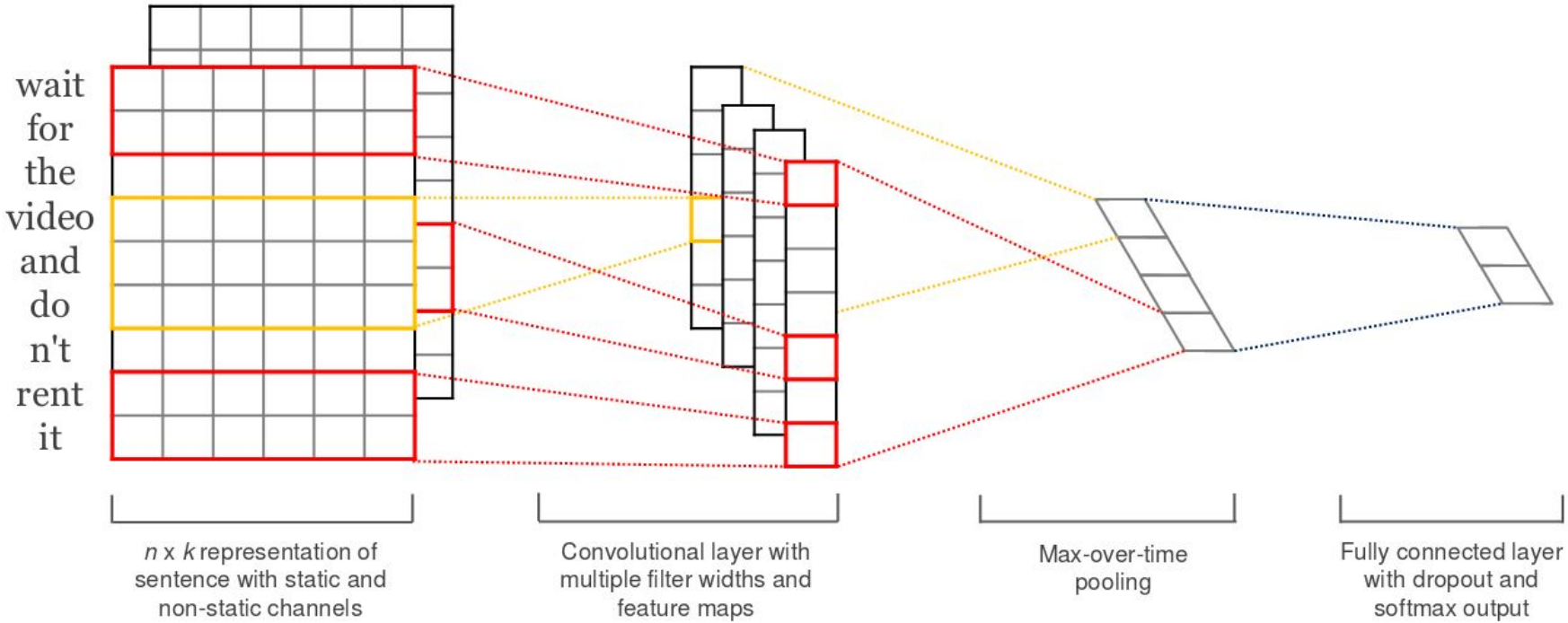


Image source: https://arxiv.org/pdf/1408.5882.pdf

Layers

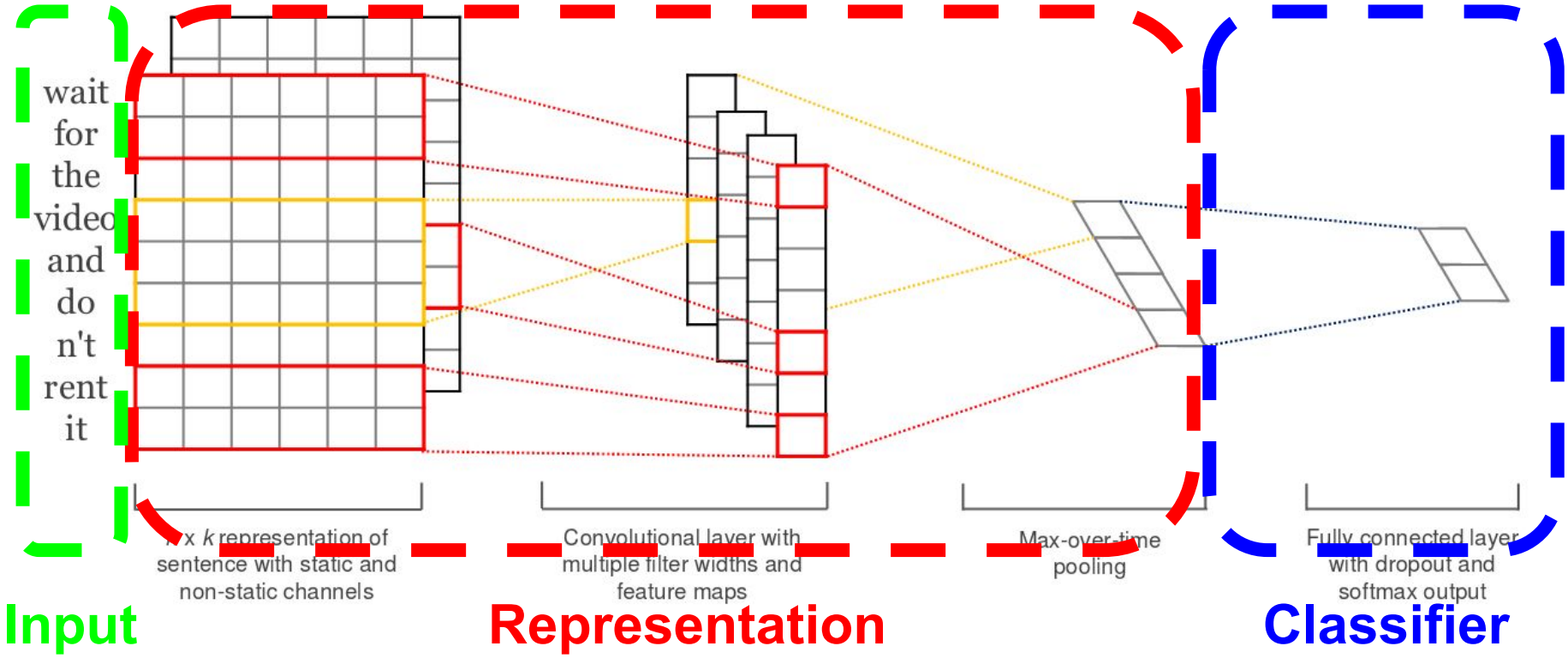


Image source: https://arxiv.org/pdf/1408.5882.pdf

NN Models

- Each NN is designed to solve a specific task
- Trained on some labeled data
- Learned representation / understanding of the input
- Learned prediction

Is this representation meaningful outside the given task?

Can we reuse it?

Image: http://themelkerproject.com/yes-we-can-binders-full-of-women-two-political-tracks-for-inauguration-day/

Transfer Learning

- Train a model on one task, re-use on a related task
- Adapt the gained knowledge
- Mostly applicable to neural networks
  - Representation learning
  - Use layers as building blocks
- Leads to
  - Faster learning
  - Better generalization

Why Transfer Learning

- Models can’t generalize well if data is not enough
- Data is never enough
- There is always bigger and better NN

Goal: Find a related task with huge dataset

Image source: https://towmuch.com/articles/the-more-the-better-the-less-the-worse-524.html

Example: Train on ImageNet

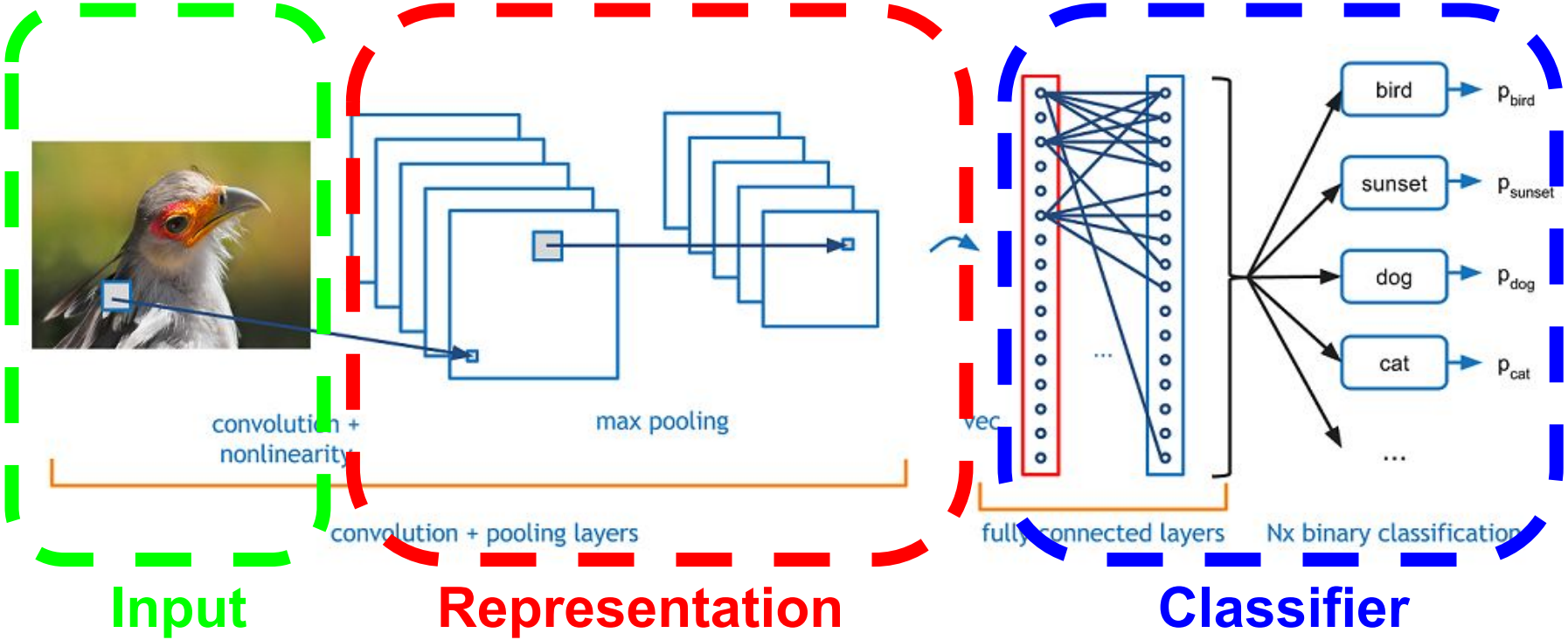
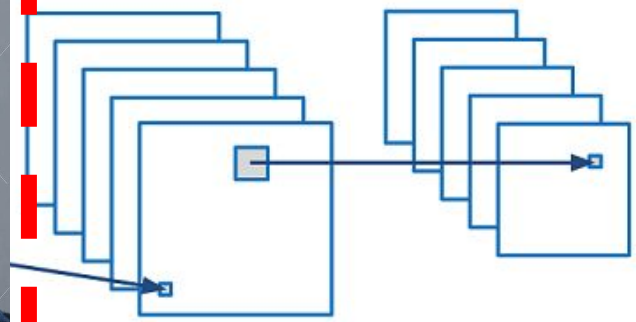
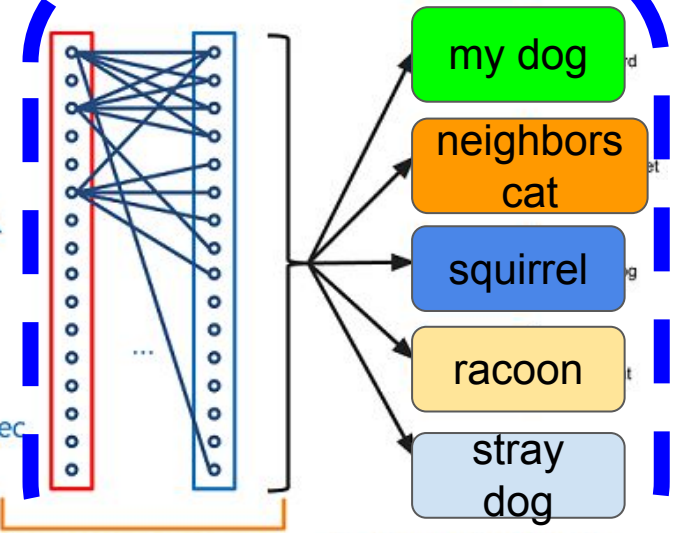


Image source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Example: Adapt to your task





convolution + nonlinearity

convolution + pooling layers

fully connected layers

Nx binary classification

Your data **Representation** **Your Classifier**

Image source: https://www.123rf.com/photo_55190461_funny-pug-dog-with-man-hands-in-checkered-shirt-pointing-on-you-and-showing-thimbs-up-over-grey-back.html

Transfer Learning as a Social Activity

- Collaborate
- Reuse models trained on large datasets
- Open source NN models

How to solve the implementation details?

Scientists' Path to Reusability - 1

- Conduct experiments, **publish a paper**
- Steps to reproduce:
  - Understand the algorithm
  - Implement from scratch
  - Guess missing implementation details
  - Find data to test and validate

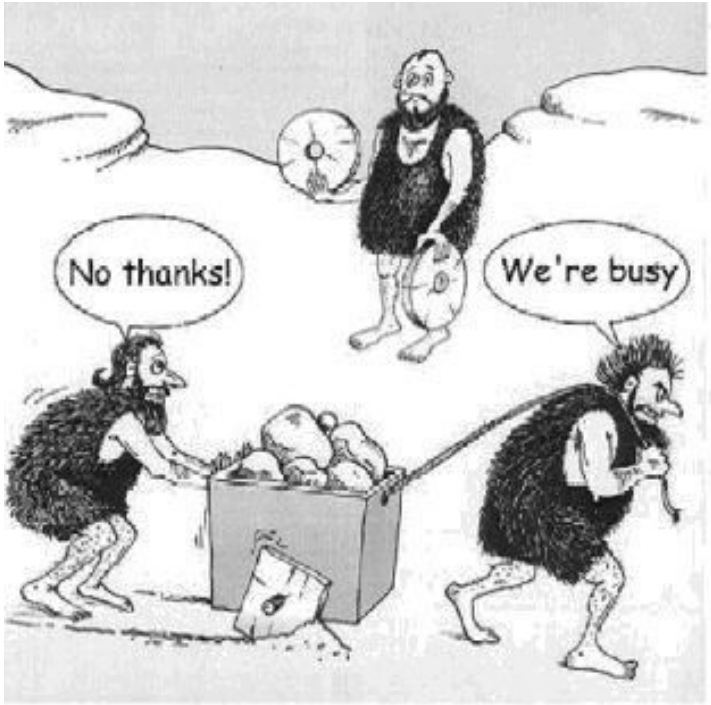


Image source: http://peterjnorth.blogspot.com/2017/05/why-are-we-reinventing-wheel.html

Scientists' Path to Reusability - 2

- Publish a paper, **share unusable code** in Matlab / C / R / Python
- Steps to reproduce:
  - Understand the algorithm
  - Implement from scratch
  - Find implementation details in code
  - Find data to test and validate

Scientists' Path to Reusability - 3

- Publish a paper, share unusable code, **shared dataset**
- Steps to reproduce:
  - Understand the algorithm
  - Implement from scratch
  - Find implementation details in code
  - Reproduce results on the dataset

- Example: A Convolutional Neural Network for Modelling Sentences, Kalchbrenner et. al, 2014, state-of-art in sentiment analysis. Code in Matlab

Scientists' Path to Reusability - 4

- Publish a paper, share **clean code using common toolkits**
- Steps to reproduce:
  - Understand the algorithm (optional)
  - Reuse code
- Reuse complex NN functions

Reusability Now

- Publish a paper
- Share clean code using common toolkits
- **Share a pretrained NN model**

Steps to reproduce:

- Run a script from a git repo

How to Use Pretrained Models

- Find a pretrained neural network on a related task
- Try the network directly on your task

OR

- Tune the network on your task (train with your data)

OR

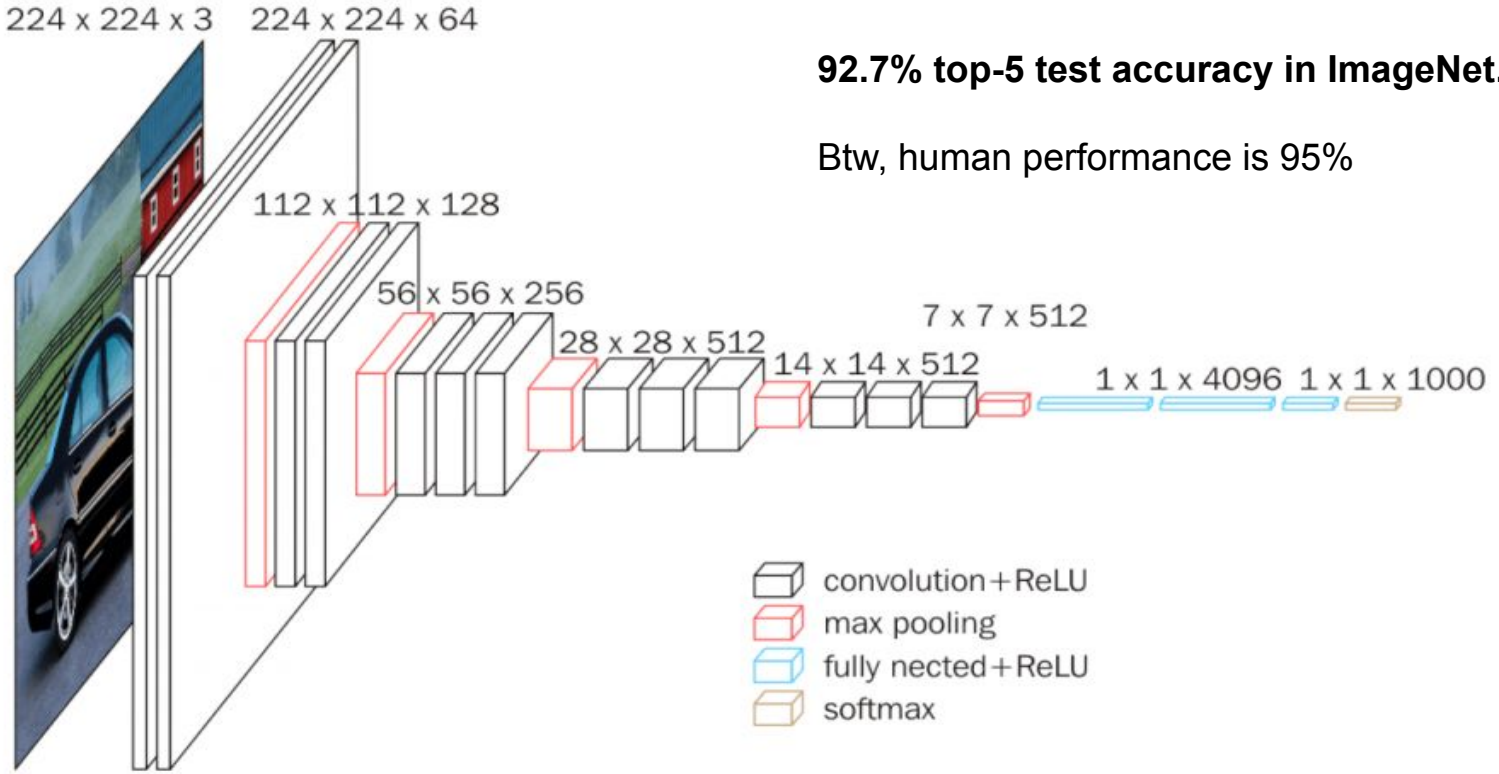
- Use some layers as representation, append your layers
- Tune the model if needed

Example in Keras

```
# example of tending the vgg16 model
from keras.applications.vgg16 import VGG16
from keras.models import Model
from keras.layers import Dense
from keras.layers import Flatten
# load model without classifier layers
model = VGG16(include_top=False, input_shape=(300, 300, 3))
# add new classifier layers
flat1 = Flatten()(model.outputs)
class1 = Dense(1024, activation='relu')(flat1)
output = Dense(10, activation='softmax')(class1)
# define new model
model = Model(inputs=model.inputs, outputs=output)
# summarize
model.summary()
# ...
```

Source: https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/

What We Created (VGG16)



92.7% top-5 test accuracy in ImageNet.

Btw, human performance is 95%

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

Latest Pretrained Models on Images

On ImageNet:

RANK	METHOD	TOP 1 ACCURACY	TOP 5 ACCURACY	NUMBER OF PARAMS	EXTRA TRAINING DATA	PAPER TITLE	YEAR
1	FixResNeXt-101 32x48d	86.4%	98.0%	829M	✓	Fixing the train-test resolution discrepancy	2019
2	ResNeXt-101 32x48d	85.4%	97.6%	829M	✓	Exploring the Limits of Weakly Supervised Pretraining	2018
3	ResNeXt-101 32x32d	85.1%	97.5%	466M	✓	Exploring the Limits of Weakly Supervised Pretraining	2018

Source: https://paperswithcode.com/sota/image-classification-on-imagenet

In Text:

- Breakthrough in NLP

- Learning a meaningless task

- Similar contexts

- Trained on billions of words

Word2vec

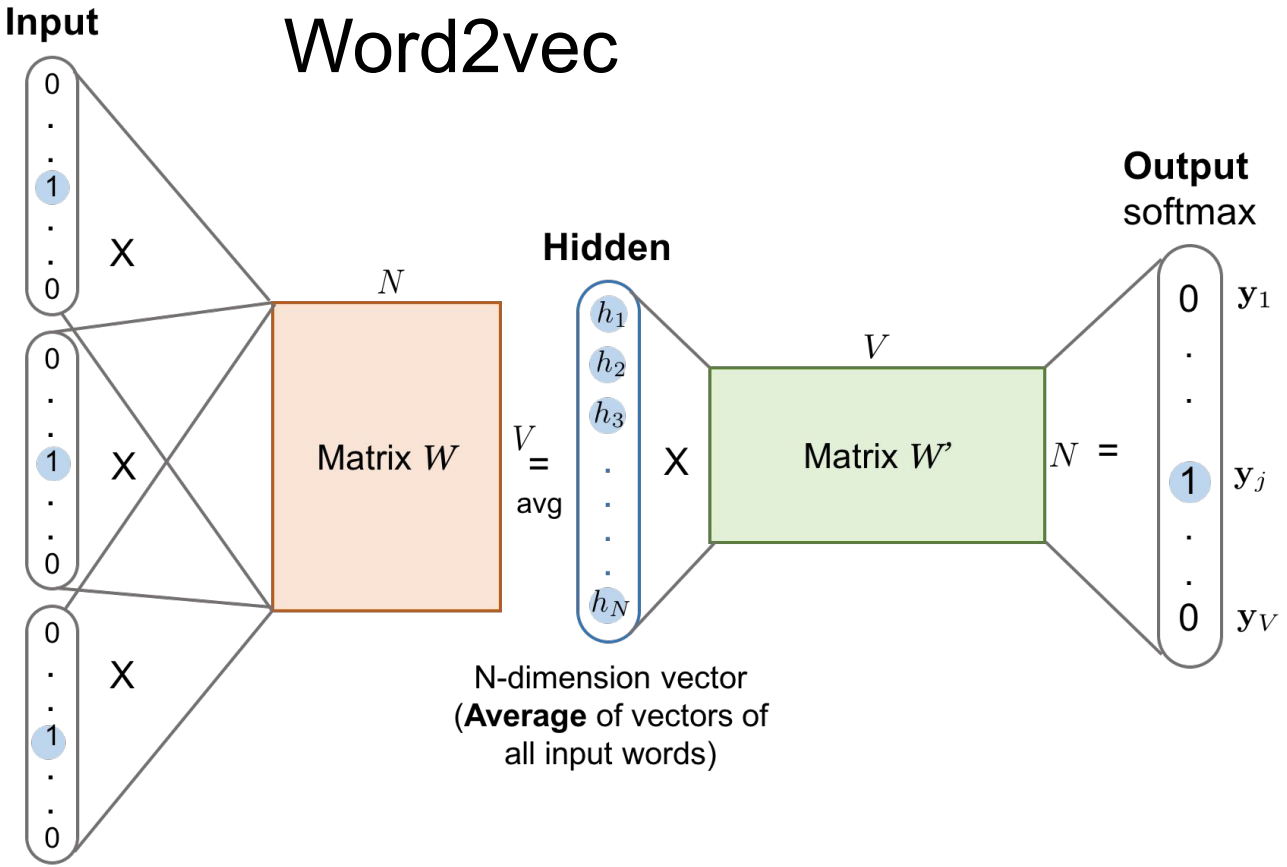


Image: https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

Word2vec

- Just reuse the weights matrix

- Third party implementations

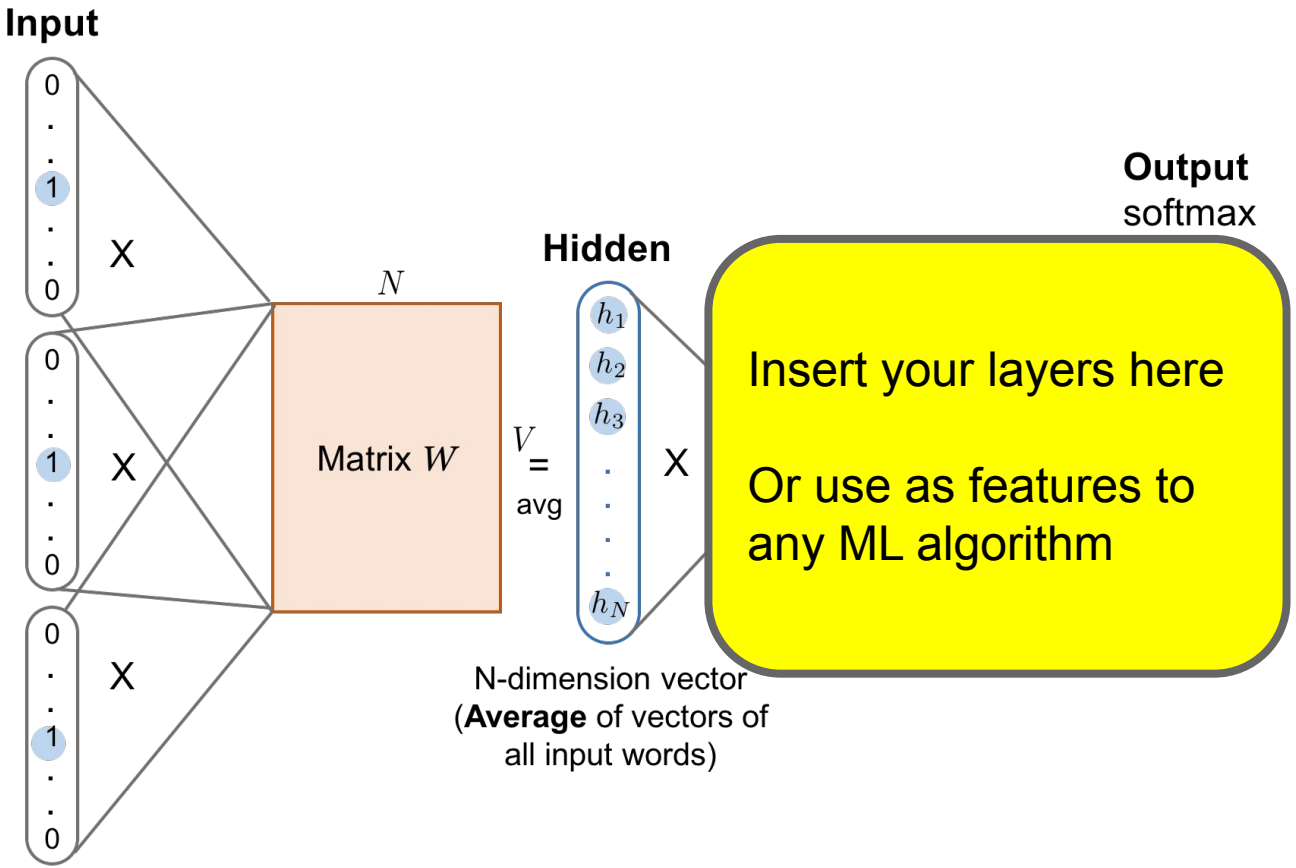


Image: https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

Pretrained Models on Text (2019)

- BERT, **340 million** params, estimated training cost: $6,912
  - State of art on 11 NLP tasks in 2019
  - Beats human performance
- Grover Mega, **1.5 billion** params, estimated training cost: $25,000
  - State of art in generating / detecting fake news
  - Beats human performance
- T5, **11 billion** parameters, estimated training cost: ??
  - Beats BERT
  - Close to **human-level language understanding**

Source: https://syncedreview.com/2019/06/27/the-staggering-cost-of-training-sota-ai-models/,

Pretrained Models on Text (2019)

- BERT
  - Bert-as-a-service (server and client)
  - Download code + models for Tensorflow and Pytorch
- Grover Mega
  - Code and model available for Tensorflow

- T5
  - Code in Tensorflow
  - Models available on GCP

	Rank	Name	Model	Score
	1	T5 Team - Google	T5	89.7
	2	ALBERT-Team Google Language	ALBERT (Ensemble)	89.4
+	3	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)	89.0
	4	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)	88.8
	5	Facebook AI	RoBERTa	88.5
	6	XLNet Team	XLNet-Large (ensemble)	88.4
+	7	Microsoft D365 AI & MSR AI	MT-DNN-ensemble	87.6
	8	GLUE Human Baselines	GLUE Human Baselines	87.1

Source: https://gluebenchmark.com/leaderboard

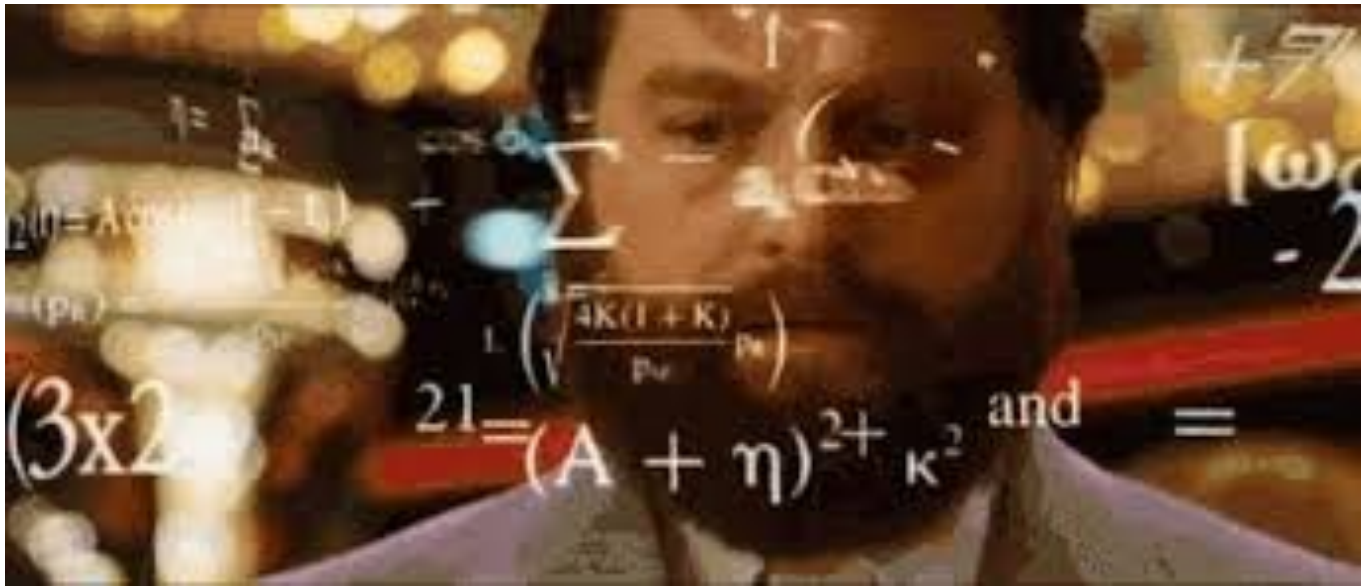
Conclusion (2019)

Non-expert humans outperform machines on GLUE* but by a small margin

** At the time of paper writing. Two systems** have since surpassed our human baseline estimate.*

*** At the time of poster creation.*

Understanding the Model (Optional)



(This often happens after the POC)

Source: https://www.inverse.com/article/56357-how-to-overcome-your-fear-of-math

BERT

- Learn embeddings
- Pre-trained on two tasks
  - Predict masked words
  - Next sentence prediction
- Data:
  - Books (800M tokens)
  - EN Wikipedia (2,5B tokens)

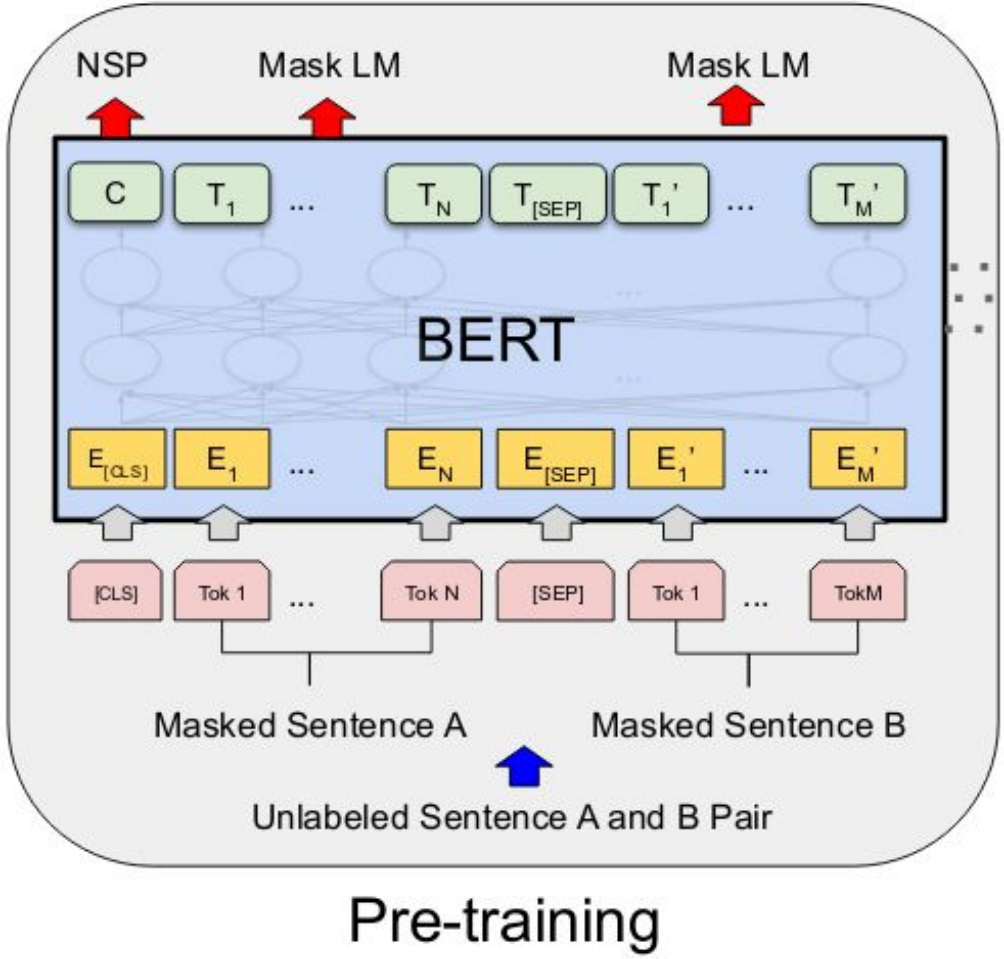


Image source: https://arxiv.org/pdf/1810.04805.pdf

BERT

- Fine-tuning on 11 NLP tasks
- Use embeddings
- Fully connected layer on top of C for classification
- Multiple models available online

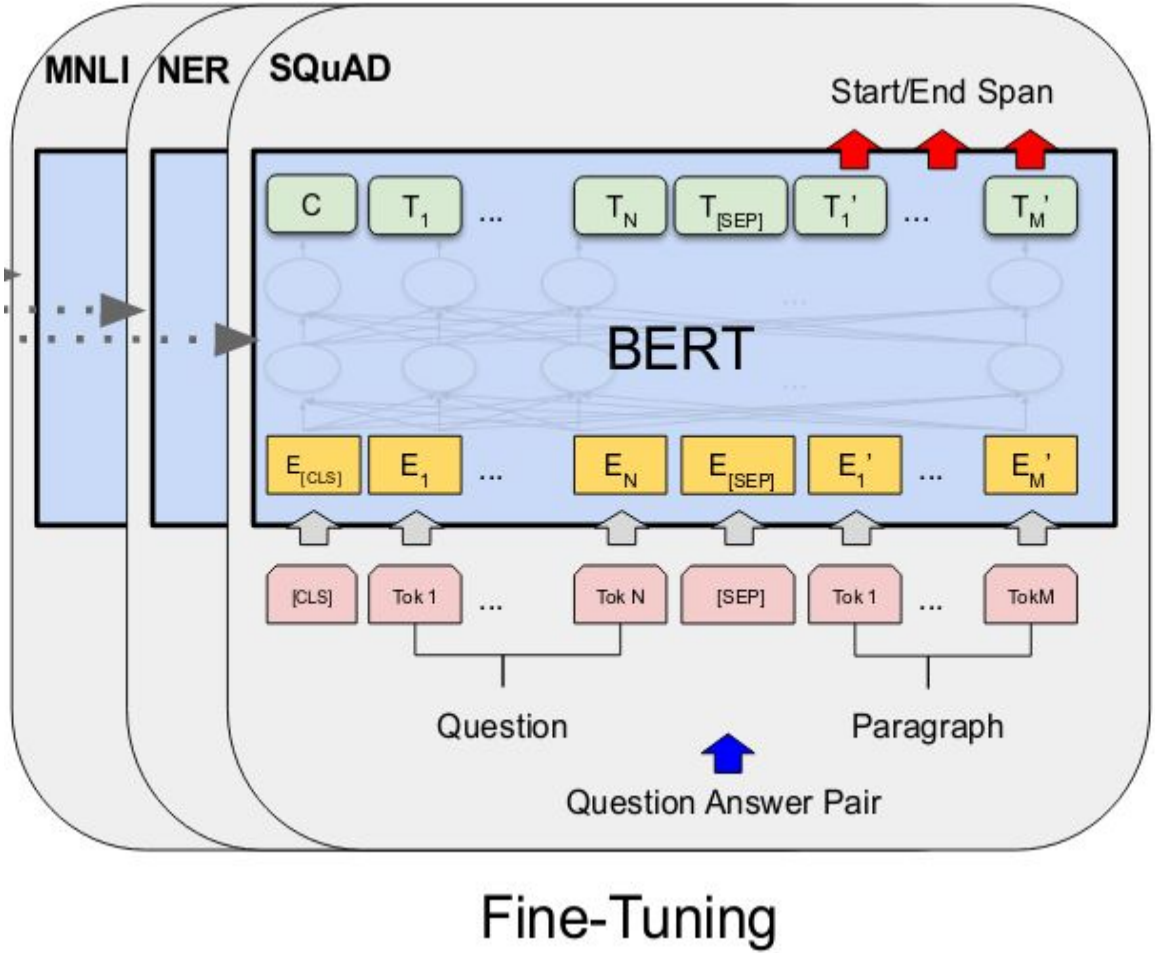


Image source: https://arxiv.org/pdf/1810.04805.pdf

Where to Start

- Python
- Tensorflow or PyTorch:
  - Open source NN frameworks
  - Latest algorithms, layers, functions
  - Large ecosystems - viral communities, toolkits
  - Production ready

Where to Get Models

- https://paperswithcode.com/
  - State-of-art work for many tasks
- https://github.com/tensorflow/models
- https://pytorch.org/hub
- https://modelzoo.co/
- https://github.com/huggingface/transformers
  - Models for NLP

Bonus: A Cool Application

- Text to Speech
- Train a large NN on a large dataset of transcripts
- (on par with humans)
- **Adapt to a specific voice using 5-10 seconds of data**
- https://www.youtube.com/watch?v=-O_hYhToKoA
- Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis, Jia et.al, 2019

THANK YOU!

Questions?