

Machine Learning in imaging and more!

22 Feb, 2018

Varun Gulshan
Research Scientist@Google Brain
[g.co/Brain](https://www.google.com/brain)

Selfie introduction



B Tech, Computer Science IIT Delhi, 2007



PhD, Computer Vision and Machine Learning, University of Oxford, 2011



Research Engineer @ Flutter (A computer vision startup, acquired by Google in 2013)



Research Scientist at Google [current]

Outline

- Deep dive into a specific Medical imaging problem (Diabetic retinopathy screening)
- Overview of Neural networks (specifically Convolutional neural network) and optimization.
- The zoo of ML models and applications

Diabetic retinopathy (DR) screening



Build an automated fundus image reading algorithm to refer 'referable' cases to an Ophthalmologist.

Fundus imaging to diagnose DR



=



DR: 5 point scale



No DR

Mild DR

Moderate DR

Severe DR

Proliferative DR

DR: 5 point scale



No DR

Mild DR

Moderate DR

Severe DR

Proliferative DR

DR: 5 point scale



No DR

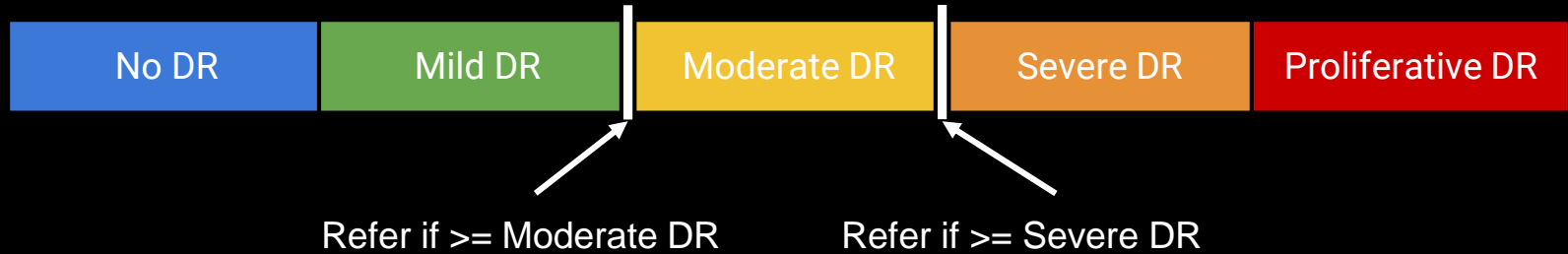
Mild DR

Moderate DR

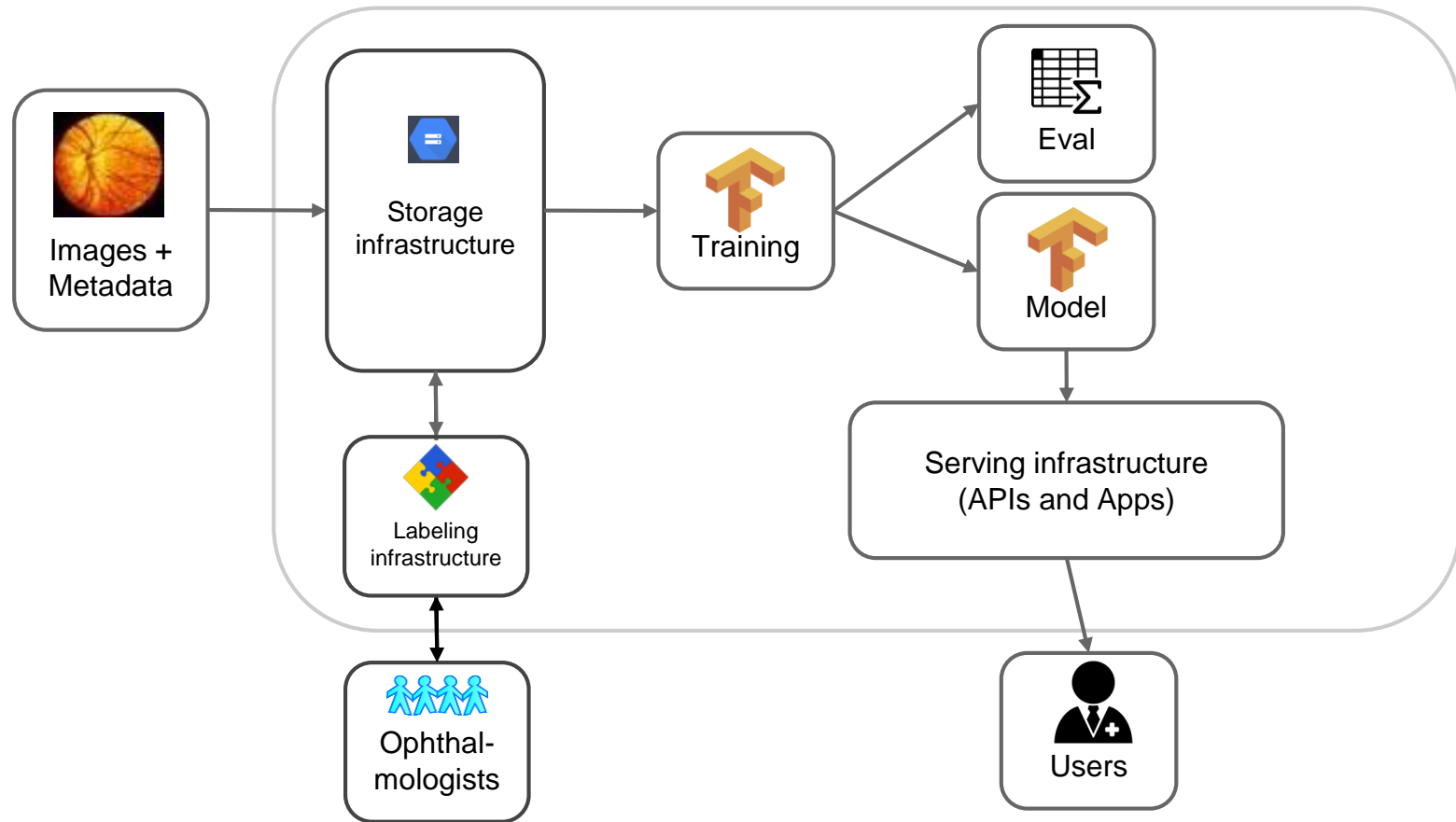
Severe DR

Proliferative DR

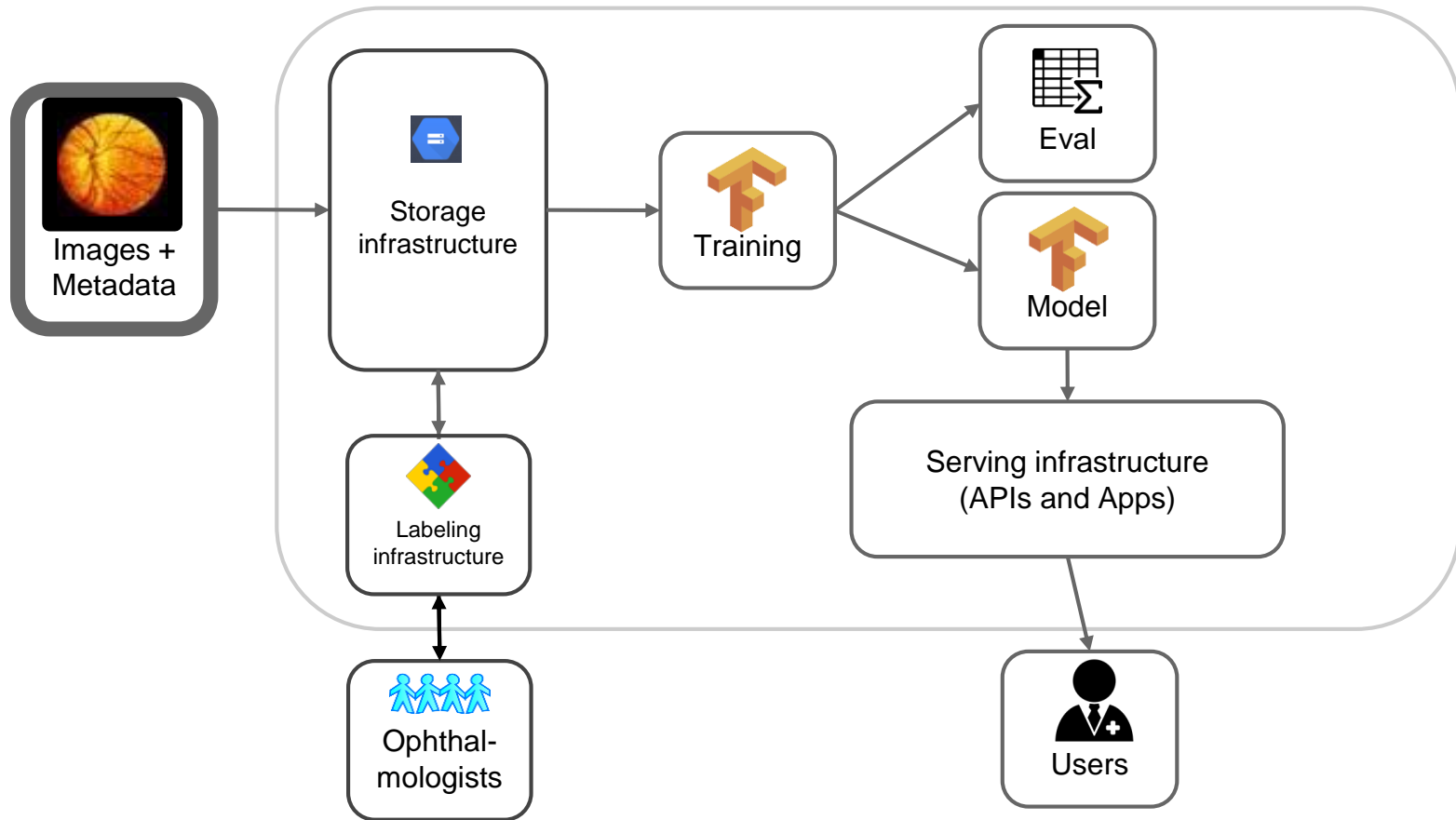
DR: Referability thresholds



Building blocks of a ML system



Building blocks of a ML system



Images and metadata

Data partners

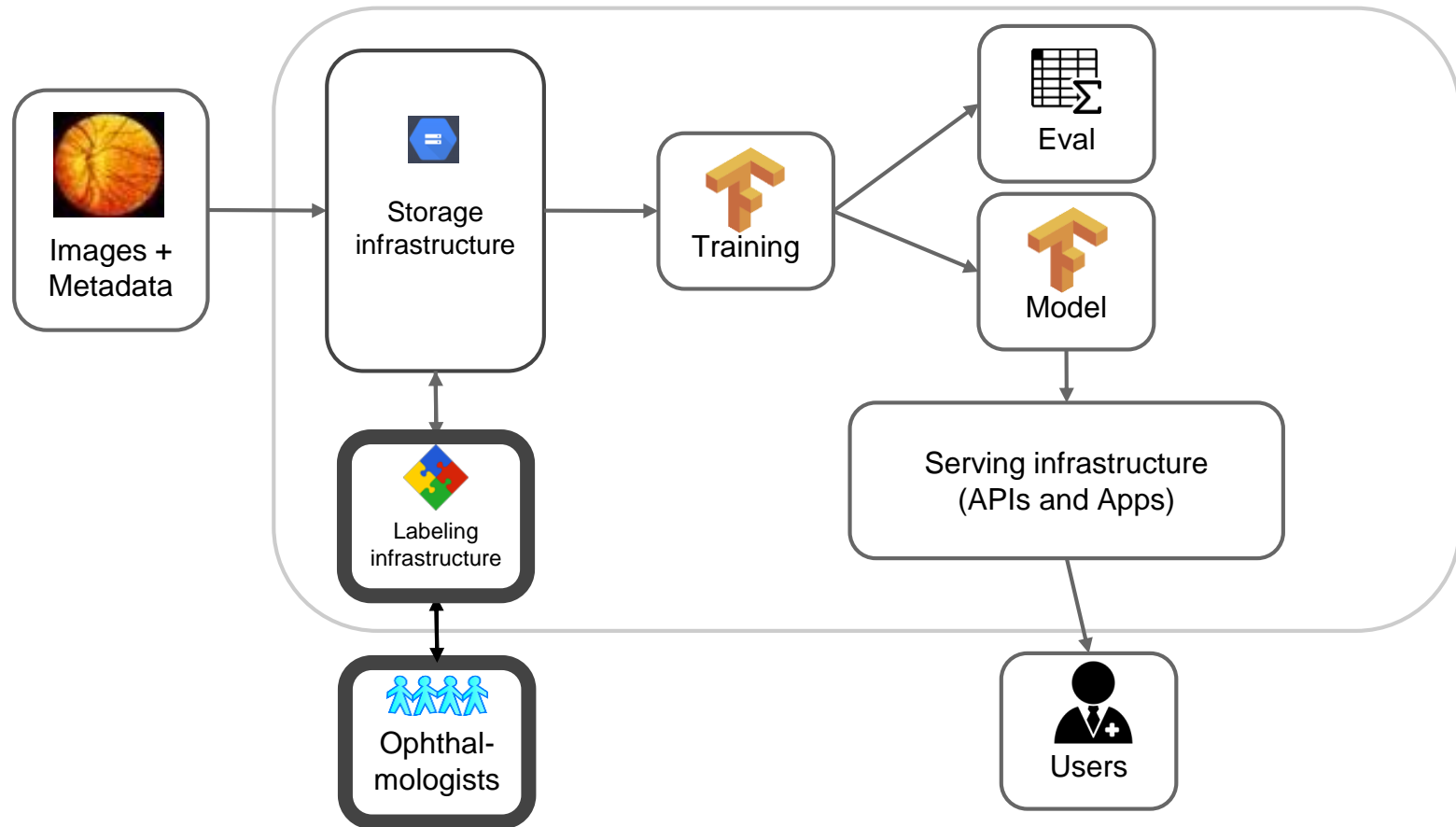


AREDS

2 Million+ images across diverse ethnicities, age groups, gender, confounding diseases.

Some come with the partners DR grading as well.

Building blocks of a ML system



Labeling tool for Ophthalmologists

Time : 0:0:52 Answered: 0

CONTRASTc

MAGNIFIERshift

DR GRADE

1NO DIABETIC RETINOPATHY

2MILD NPDR

3MODERATE NPDR

4SEVERE NPDR

5PROLIFERATIVE DR

REFERABLE DME? (Q)

NO

YES

PRP LASER SCARS? (A)

NO

YES


FOCAL LASER SCARS? (S)

NO

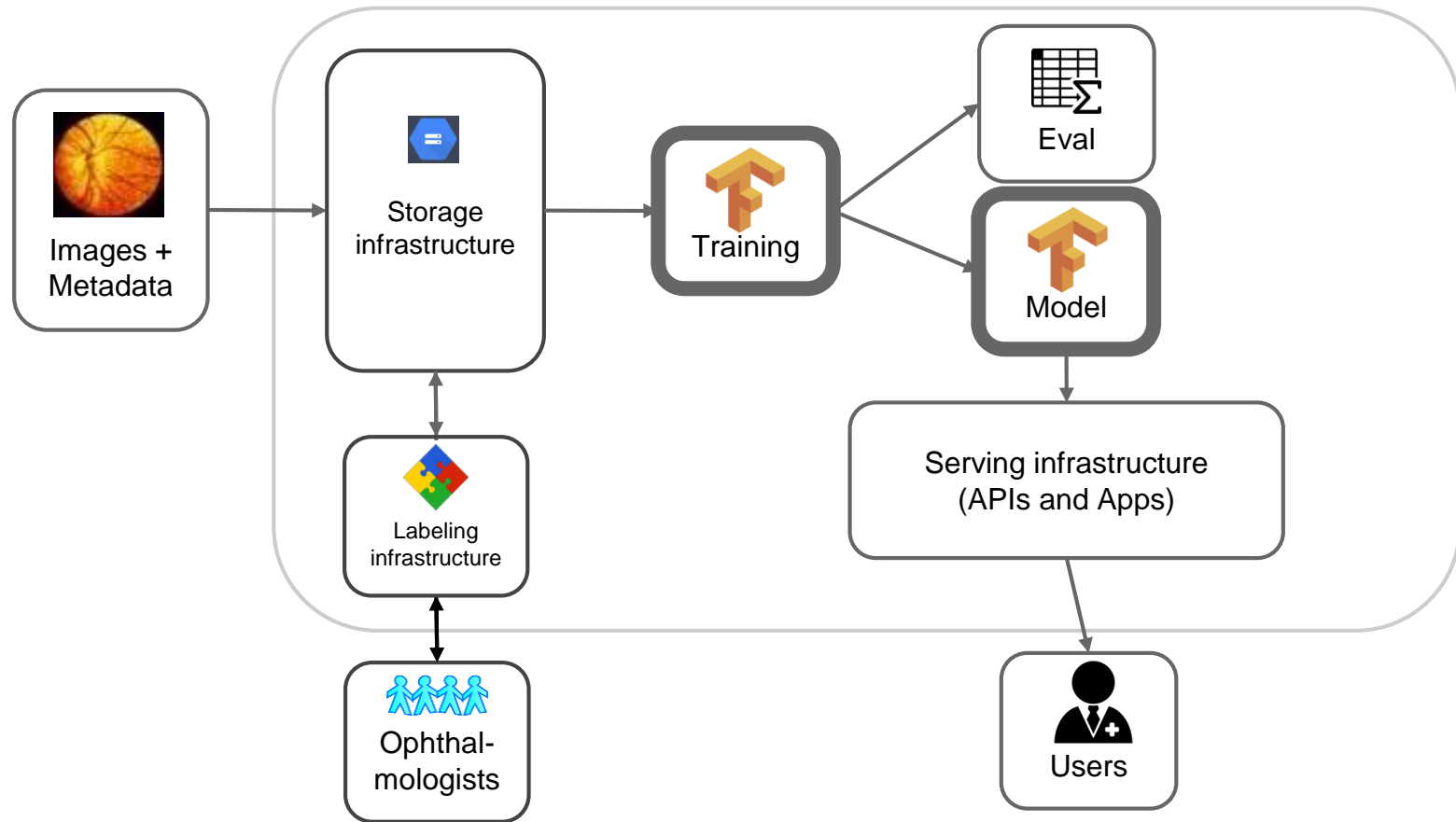
YES

BACK

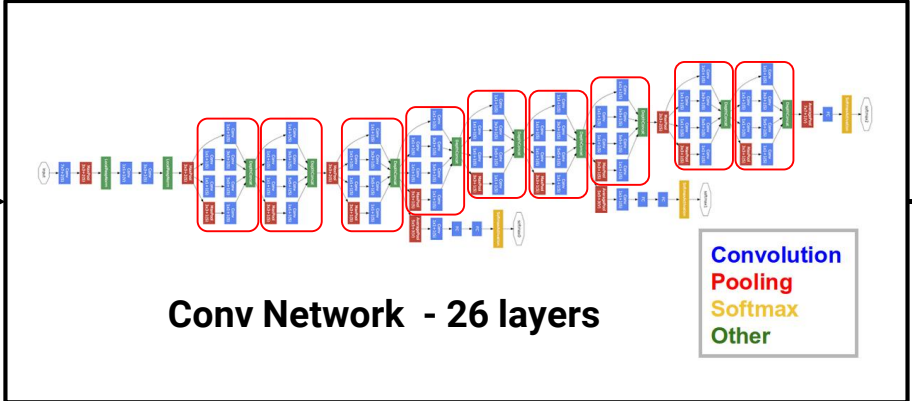
NEXT



Building blocks of a ML system



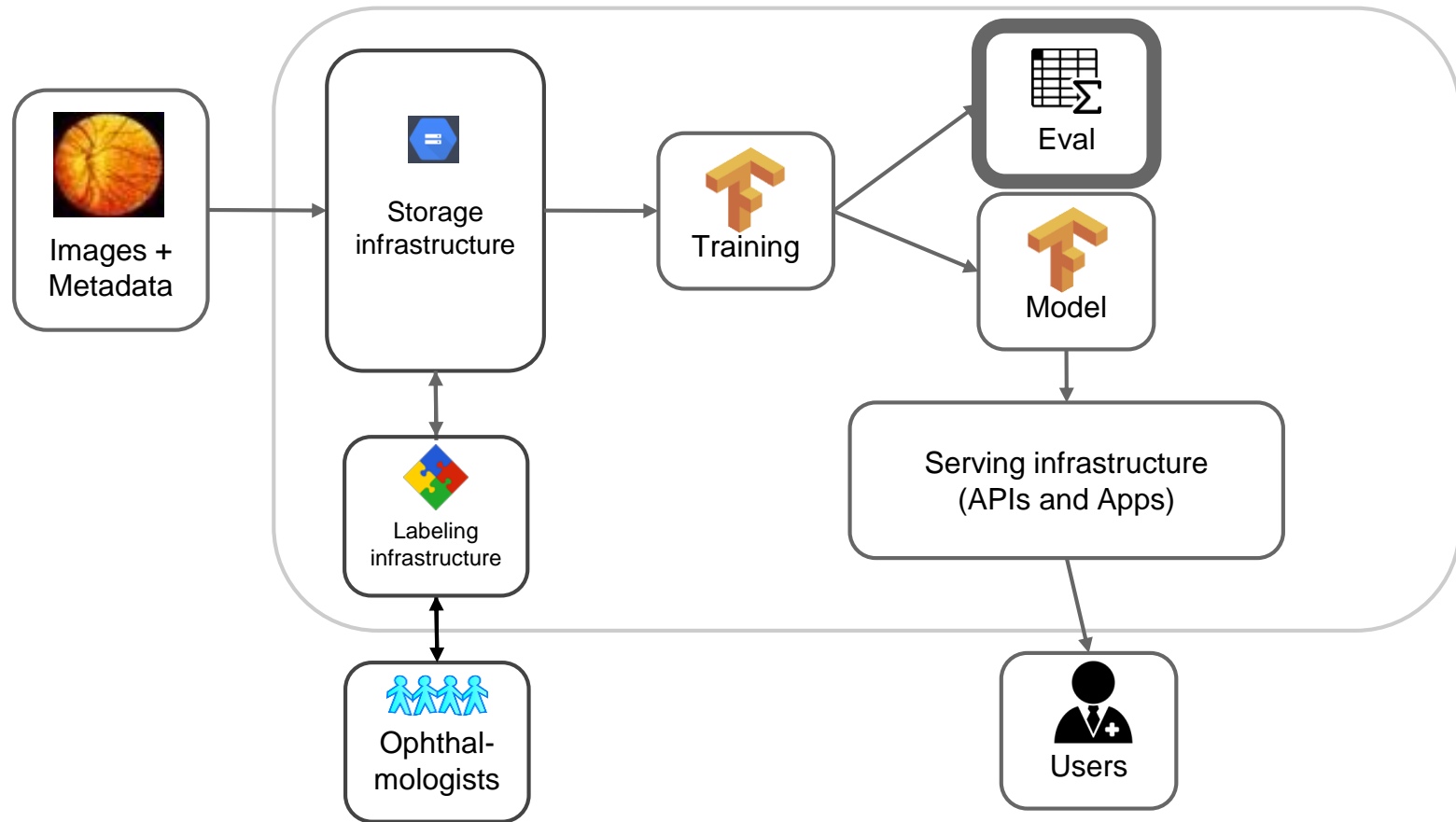
Model: Deep neural networks



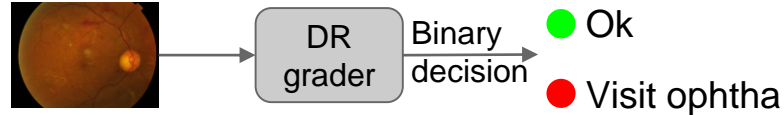
GoogLeNet

- Image Quality
- DR Severity
- DME Grade
- Gender
- Left/Right

Building blocks of a ML system

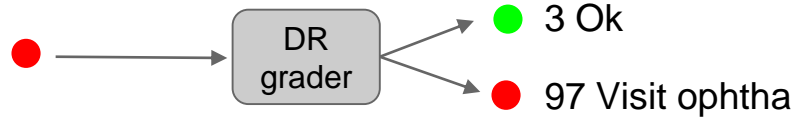


Measuring performance: Sensitivity and specificity



Sensitivity

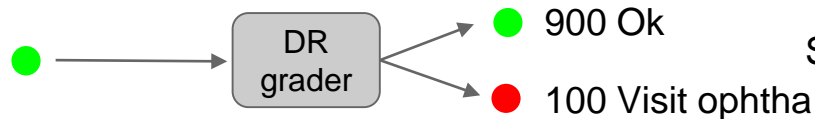
100 referable patients



$$\text{Sensitivity} = \frac{\text{True positives}}{\text{Total positives}} = 97\%$$

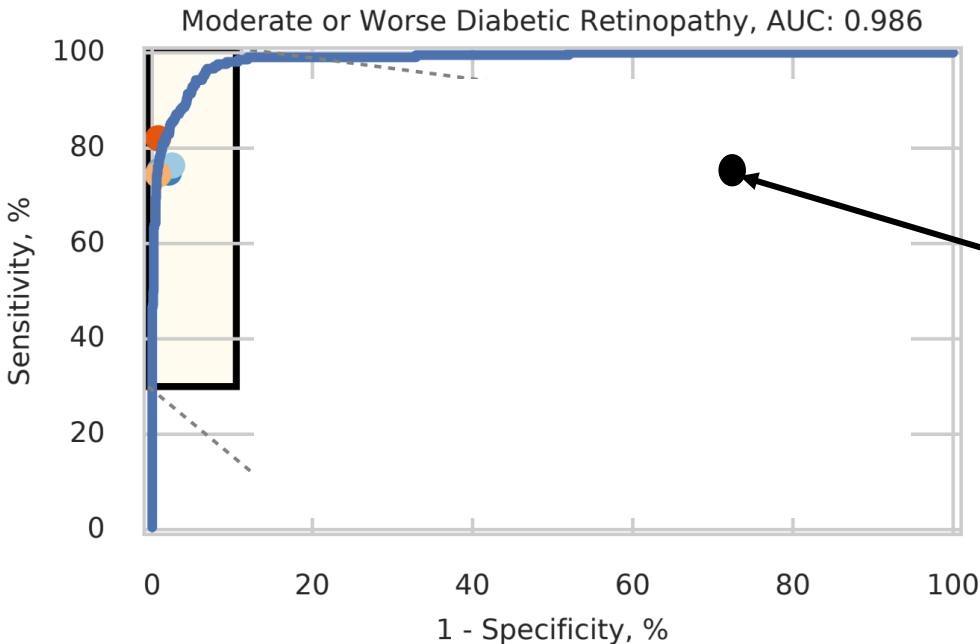
Specificity

1000 healthy patients



$$\text{Specificity} = \frac{\text{True negatives}}{\text{Total negatives}} = 90\%$$

ROC curves



	DR	
	Sensitivity	Specificity
Algorithm	97.1%	92.3%

Full paper at: <https://arxiv.org/pdf/1710.01711.pdf> [Published in Ophthalmology]

Improvement of our original work published in JAMA: <https://research.google.com/pubs/pub45732.html>

Drag another image to analyze, or

CHOOSE IMAGE

FILENAME (SIZE)

07.jpg (276 KB)

DIABETIC RETINOPATHY GRADE

1.7 (Mild)

DIAGNOSIS ID

028371031

Recent Images

FILENAME / DR GRADE

07.jpg / 2.6



FILENAME / DR GRADE

06.jpg / 3.2



FILENAME / DR GRADE

05.jpg / 2.3



FILENAME / DR GRADE

04.jpg / 2.6



FILENAME / DR GRADE

03.jpg / 1.6



FILENAME / DR GRADE

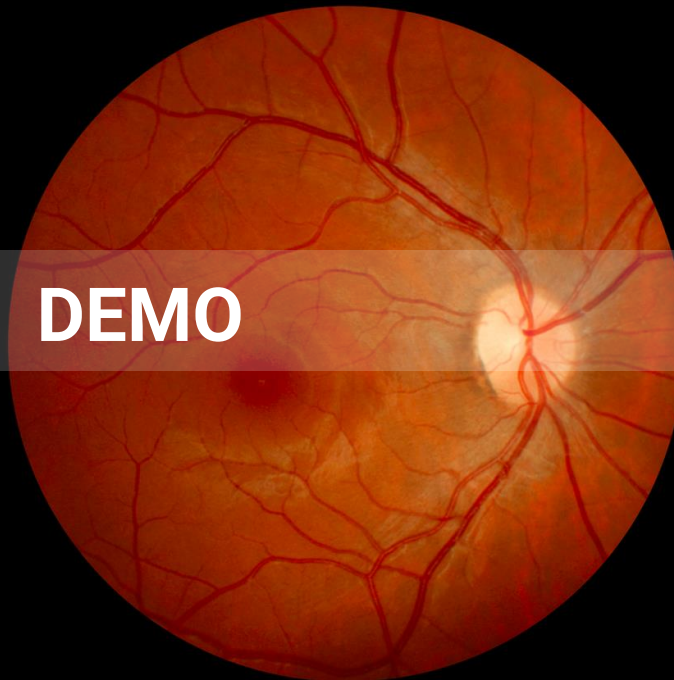
02.jpg / 0.9



FILENAME / DR GRADE



DEMO



Outline

- Deep dive into a specific Medical imaging problem (Diabetic retinopathy screening)
- **Overview of Neural networks (specifically Convolutional neural network) and optimization.**
- The zoo of ML models and applications

The function we want to learn



Referable DR = Yes/No

Input representation

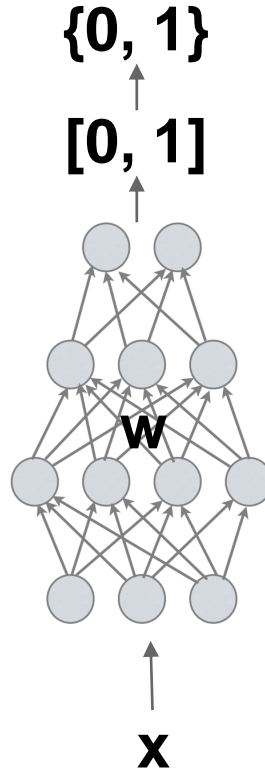
31	29	22	15	10	30	24	18	25	41	25	14	32	38	42	45	68	85	76	61	68	66	57	43	35	29	30	45	33	18	14	23	47	68	73	51	37	36	27	27
22	22	22	15	14	31	24	24	15	35	43	18	23	37	45	51	58	74	71	61	67	66	52	35	32	28	27	35	28	18	15	23	42	68	67	36	29	31	24	24
21	27	25	10	14	40	32	21	12	29	46	29	29	30	53	61	50	59	62	57	56	55	46	34	31	25	20	24	23	19	22	29	37	62	60	32	28	30	27	28
17	25	28	23	11	34	33	18	14	32	51	55	52	25	46	60	45	52	65	53	37	40	41	33	31	26	18	18	19	21	35	42	43	82	54	30	28	31	35	35
13	12	18	28	20	14	24	26	29	31	58	65	68	50	38	40	40	52	82	58	41	40	38	28	28	29	20	15	17	24	45	58	57	63	47	38	38	34	39	35
13	15	21	24	30	14	34	40	29	29	58	82	74	63	82	38	48	82	58	55	44	40	38	34	27	28	21	18	20	26	43	54	57	58	44	37	37	41	40	31
12	8	19	33	38	26	28	40	33	30	42	58	73	58	102	95	72	71	78	77	38	32	37	44	32	24	18	18	21	24	31	34	36	41	47	48	58	44	34	24
8	3	13	32	38	38	38	38	29	32	41	57	72	65	60	121	113	187	132	115	35	21	23	30	24	17	13	14	20	18	22	24	28	41	57	64	58	48	28	28
5	2	8	28	38	30	40	41	28	30	42	56	55	61	60	63	69	88	123	139	44	29	17	19	17	12	8	15	21	21	29	34	39	60	63	89	67	36	30	28
2	1	2	11	18	33	31	34	29	29	27	33	38	12	33	22	31	77	95	89	59	21	19	20	24	21	11	17	21	25	43	58	58	82	99	102	74	40	35	31
1	1	2	9	16	32	25	29	18	34	38	32	31	2	3	7	18	38	61	68	118	32	18	21	38	35	16	17	22	29	58	88	88	101	97	101	98	59	48	31
2	1	2	7	12	18	24	23	18	37	30	22	24	4	2	8	15	22	35	112	125	35	23	21	30	29	14	18	25	37	51	88	91	183	87	93	93	76	55	36
17	5	2	6	7	14	20	29	25	35	25	26	25	3	6	21	37	44	82	185	88	23	21	18	18	17	13	16	34	48	51	58	88	91	73	77	78	88	65	42
22	13	5	8	5	8	14	20	27	33	24	35	25	4	8	31	84	123	161	167	38	21	17	13	14	15	17	26	48	58	58	88	84	75	82	65	74	74	63	45
10	11	7	9	8	11	17	30	55	60	41	28	18	12	11	15	82	188	228	88	19	25	18	13	13	19	29	43	54	68	71	88	84	63	51	61	72	67	55	42
11	8	9	15	8	16	34	64	62	41	32	30	38	17	15	17	24	88	295	64	21	27	20	15	29	33	48	58	59	65	84	95	88	58	52	63	68	57	48	35
25	7	7	14	8	9	44	42	25	24	32	37	48	32	28	18	21	58	171	89	23	26	28	18	31	58	64	67	88	65	63	67	72	55	56	67	62	51	45	41
35	7	4	8	8	26	37	21	18	28	28	38	54	47	24	25	25	45	155	133	25	24	18	25	43	58	68	64	52	57	72	78	87	54	53	65	61	52	47	58
31	11	3	7	16	27	19	18	18	15	24	38	59	71	48	38	54	85	184	137	28	23	22	38	48	57	67	64	41	58	67	73	65	58	53	58	58	55	51	53
23	17	3	7	15	15	11	13	14	15	21	38	85	86	118	58	58	141	178	73	23	26	31	43	45	51	61	48	47	88	75	88	71	88	48	48	58	54	47	52
18	27	4	4	7	5	4	8	18	14	20	28	39	72	125	98	58	127	183	34	28	31	38	44	48	48	81	58	65	73	82	85	75	88	43	48	61	45	34	45
11	28	22	11	8	1	1	5	7	8	10	15	22	47	103	127	93	128	43	35	32	38	38	38	43	58	72	71	88	72	85	81	85	54	48	58	88	37	28	33
13	7	14	18	7	0	1	3	5	5	7	14	22	34	78	138	188	81	42	48	43	45	48	38	58	67	72	78	65	88	84	74	58	54	38	44	55	38	25	33
9	4	4	8	4	0	1	2	4	5	9	12	18	28	84	157	111	84	38	45	54	51	45	48	68	64	88	63	61	62	77	88	55	54	33	36	48	32	21	34
23	4	3	8	3	0	1	1	4	5	11	14	16	29	88	182	117	89	43	58	63	55	48	58	65	88	58	58	57	58	72	55	38	38	28	35	42	22	18	31
28	6	2	5	6	1	0	1	3	8	13	19	24	58	117	123	188	185	81	82	62	55	55	58	88	74	63	58	54	56	61	44	29	31	29	31	31	18	18	38
12	3	8	3	11	3	0	0	0	3	9	17	28	48	82	94	98	188	182	61	58	53	53	57	65	71	55	45	48	48	41	32	27	33	32	28	28	14	12	28
12	7	13	22	32	32	31	29	26	18	13	18	9	15	27	44	55	88	83	52	58	56	52	54	88	58	43	38	48	48	38	31	32	37	34	38	21	12	9	18
8	7	11	22	28	88	188	88	115	84	81	67	15	8	18	38	78	188	113	75	57	58	57	58	53	48	34	38	58	51	43	41	48	48	48	38	28	14	18	18

Referable DR = Yes/No

$$f \left(\begin{matrix} 31 & 29 & 22 & 15 & 19 & 30 & 24 & 18 & 25 & 41 & 25 & 14 & 32 & 38 & 42 & 45 & 66 & 65 & 70 & 61 & 68 & 68 & 57 & 43 & 35 & 29 & 30 & 45 & 33 & 18 & 14 & 23 & 47 & 60 & 73 & 51 & 37 & 38 & 27 & 27 \\ 22 & 22 & 22 & 15 & 14 & 31 & 24 & 24 & 15 & 35 & 43 & 10 & 23 & 37 & 45 & 51 & 50 & 74 & 71 & 61 & 67 & 68 & 52 & 35 & 32 & 28 & 27 & 35 & 28 & 18 & 15 & 23 & 42 & 60 & 67 & 38 & 29 & 31 & 24 & 24 \\ 21 & 27 & 25 & 10 & 14 & 40 & 32 & 21 & 12 & 28 & 48 & 29 & 29 & 30 & 53 & 61 & 50 & 50 & 62 & 57 & 56 & 55 & 46 & 34 & 31 & 25 & 20 & 24 & 23 & 19 & 22 & 29 & 37 & 62 & 60 & 32 & 28 & 30 & 27 & 28 \\ 17 & 25 & 28 & 23 & 11 & 34 & 33 & 18 & 14 & 32 & 51 & 55 & 52 & 25 & 40 & 60 & 45 & 52 & 65 & 53 & 37 & 40 & 41 & 33 & 31 & 28 & 18 & 18 & 19 & 21 & 35 & 42 & 43 & 62 & 54 & 30 & 28 & 31 & 35 & 35 \\ 13 & 12 & 18 & 28 & 20 & 14 & 24 & 26 & 20 & 31 & 58 & 65 & 60 & 50 & 38 & 40 & 40 & 52 & 62 & 58 & 41 & 40 & 38 & 28 & 28 & 20 & 15 & 17 & 24 & 45 & 56 & 57 & 63 & 47 & 38 & 30 & 34 & 39 & 35 \\ 13 & 15 & 21 & 24 & 30 & 14 & 34 & 40 & 29 & 29 & 50 & 62 & 74 & 83 & 62 & 38 & 48 & 62 & 58 & 55 & 44 & 40 & 38 & 34 & 27 & 26 & 21 & 19 & 20 & 26 & 43 & 54 & 57 & 56 & 44 & 37 & 37 & 41 & 40 & 31 \\ 12 & 8 & 19 & 33 & 38 & 26 & 28 & 40 & 33 & 30 & 42 & 59 & 70 & 56 & 102 & 95 & 72 & 71 & 70 & 77 & 38 & 32 & 37 & 44 & 32 & 24 & 18 & 18 & 21 & 24 & 31 & 34 & 36 & 41 & 47 & 48 & 50 & 44 & 34 & 24 \\ 8 & 3 & 13 & 32 & 30 & 33 & 36 & 38 & 29 & 32 & 41 & 57 & 72 & 65 & 60 & 121 & 113 & 107 & 132 & 115 & 35 & 21 & 23 & 30 & 24 & 17 & 13 & 14 & 20 & 19 & 22 & 24 & 26 & 41 & 57 & 64 & 50 & 40 & 26 & 20 \\ 5 & 2 & 8 & 29 & 38 & 30 & 40 & 41 & 28 & 30 & 42 & 56 & 55 & 61 & 60 & 63 & 69 & 68 & 123 & 139 & 44 & 29 & 17 & 19 & 17 & 12 & 9 & 15 & 21 & 21 & 29 & 34 & 39 & 60 & 83 & 89 & 67 & 36 & 30 & 26 \\ 2 & 1 & 2 & 11 & 18 & 33 & 31 & 34 & 29 & 29 & 27 & 33 & 30 & 12 & 33 & 22 & 31 & 77 & 95 & 89 & 59 & 21 & 19 & 20 & 24 & 21 & 11 & 17 & 21 & 25 & 43 & 50 & 58 & 62 & 99 & 102 & 74 & 40 & 35 & 31 \\ 1 & 1 & 2 & 9 & 10 & 32 & 25 & 28 & 16 & 34 & 30 & 32 & 31 & 2 & 3 & 7 & 18 & 38 & 61 & 68 & 118 & 32 & 18 & 21 & 36 & 35 & 16 & 17 & 22 & 29 & 58 & 60 & 80 & 191 & 87 & 191 & 90 & 59 & 40 & 31 \\ 2 & 1 & 2 & 7 & 12 & 18 & 24 & 23 & 19 & 37 & 30 & 22 & 24 & 4 & 2 & 6 & 15 & 22 & 35 & 112 & 125 & 35 & 23 & 21 & 30 & 29 & 14 & 18 & 25 & 37 & 61 & 60 & 91 & 163 & 67 & 63 & 93 & 78 & 55 & 38 \\ 17 & 5 & 2 & 6 & 7 & 14 & 20 & 29 & 25 & 35 & 25 & 26 & 26 & 3 & 6 & 21 & 37 & 44 & 82 & 105 & 80 & 23 & 21 & 18 & 18 & 17 & 13 & 16 & 34 & 40 & 51 & 59 & 68 & 91 & 73 & 77 & 70 & 80 & 65 & 42 \\ 22 & 13 & 5 & 0 & 5 & 8 & 14 & 20 & 27 & 33 & 24 & 35 & 25 & 4 & 8 & 31 & 64 & 123 & 181 & 271 & 29 & 21 & 17 & 13 & 14 & 15 & 17 & 26 & 46 & 56 & 58 & 68 & 84 & 75 & 62 & 65 & 74 & 74 & 63 & 45 \\ 19 & 11 & 7 & 9 & 8 & 11 & 17 & 38 & 65 & 60 & 41 & 29 & 10 & 12 & 11 & 15 & 62 & 100 & 228 & 169 & 25 & 18 & 13 &td="10">23 & 19 & 29 & 43 & 64 & 60 & 71 & 68 & 84 & 83 & 51 & 61 & 72 & 67 & 55 & 42 \\ 11 & 9 & 9 & 15 & 8 & 10 & 34 & 64 & 62 & 41 & 32 & 30 & 30 & 17 & 15 & 17 & 24 & 88 & 205 & 64 & 21 & 27 & 20 & 15 & 20 & 33 & 40 & 59 &td="10">59 & 65 & 64 & 95 &td="10">60 & 50 &td="10">52 &td="10">63 &td="10">68 &td="10">57 &td="10">48 &td="10">36 \\ 25 & 7 & 7 & 14 & 8 & 9 & 44 & 42 & 25 & 24 & 32 & 37 &td="10">48 &td="10">32 &td="10">20 &td="10">19 &td="10">21 &td="10">58 &td="10">171 &td="10">80 &td="10">23 &td="10">28 &td="10">20 &td="10">18 &td="10">31 &td="10">50 &td="10">64 &td="10">67 &td="10">68 &td="10">65 &td="10">83 &td="10">87 &td="10">72 &td="10">55 &td="10">56 &td="10">67 &td="10">62 &td="10">51 &td="10">45 &td="1$$

Neural networks learn this complex function

$$f\left(\begin{array}{c} \text{[grid of dots]} \\ \mathbf{x} \end{array}; \mathbf{w}\right) =$$

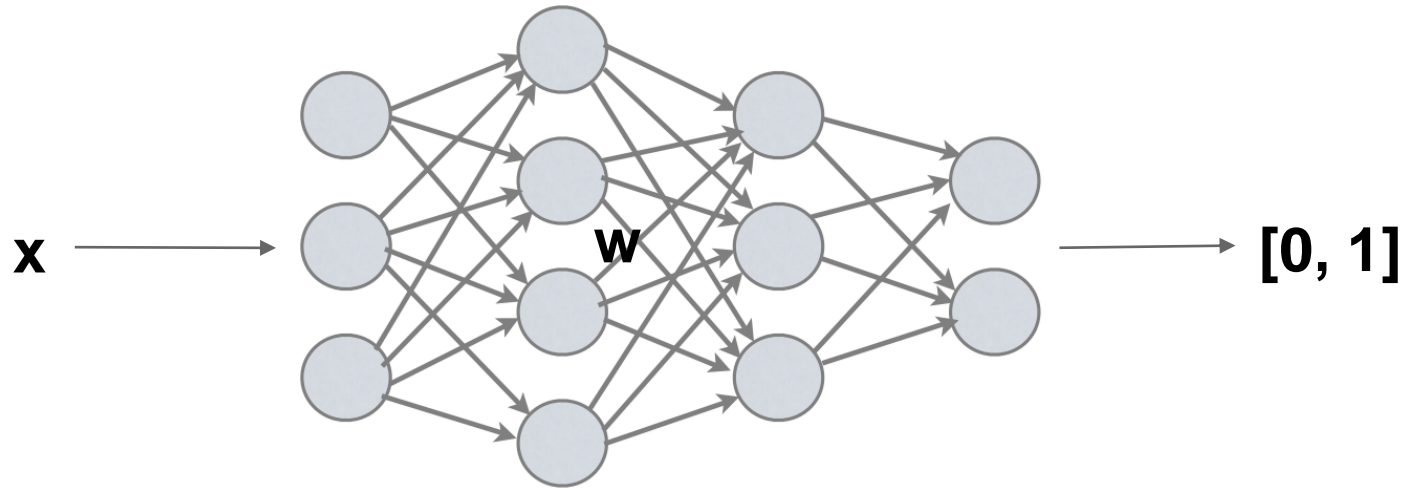


Feed raw pixels \mathbf{x} into model

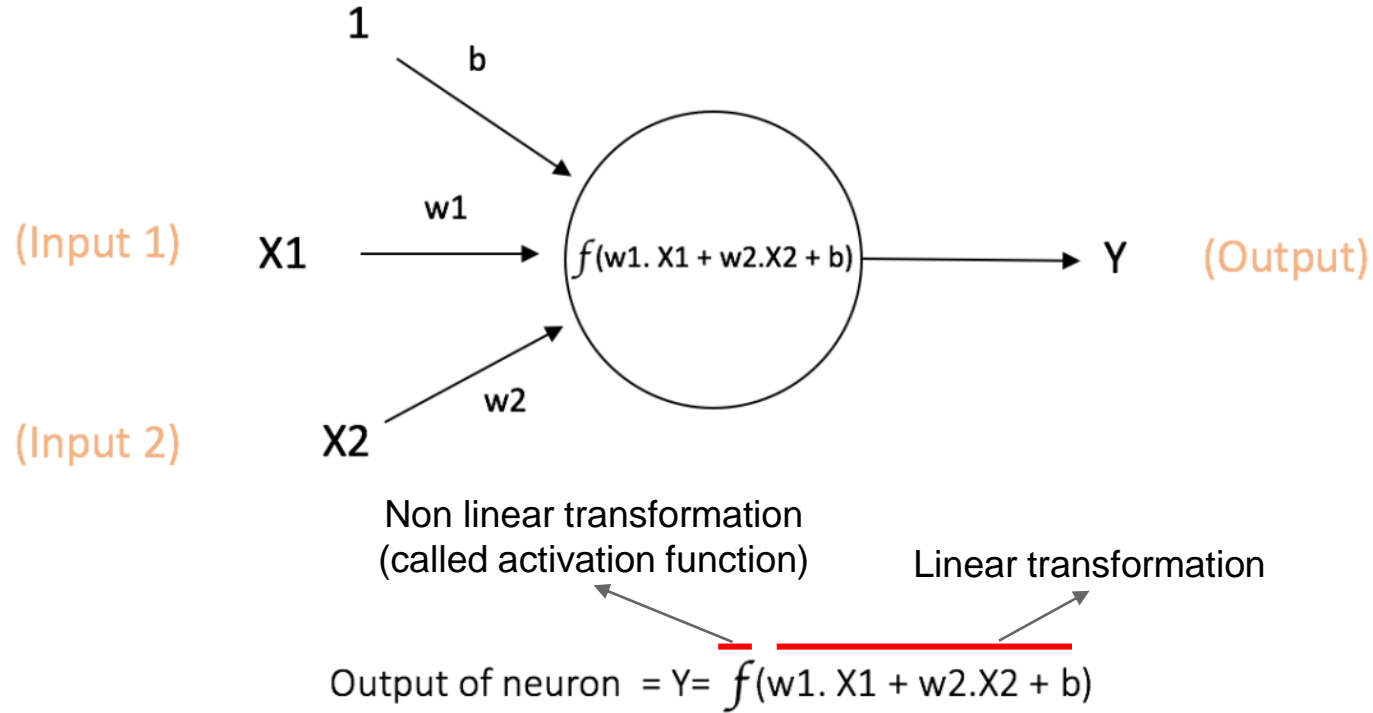
Learn good values for \mathbf{w} over training data

Have to choose good network architectures

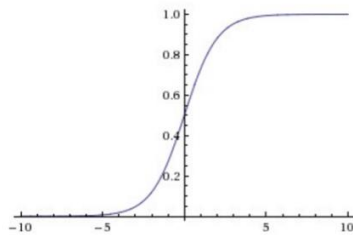
What is inside the neural network?



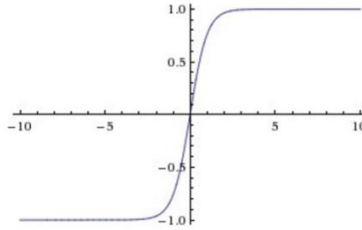
A single neuron



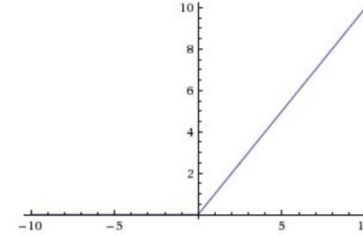
Some commonly used activation functions



Sigmoid



tanh

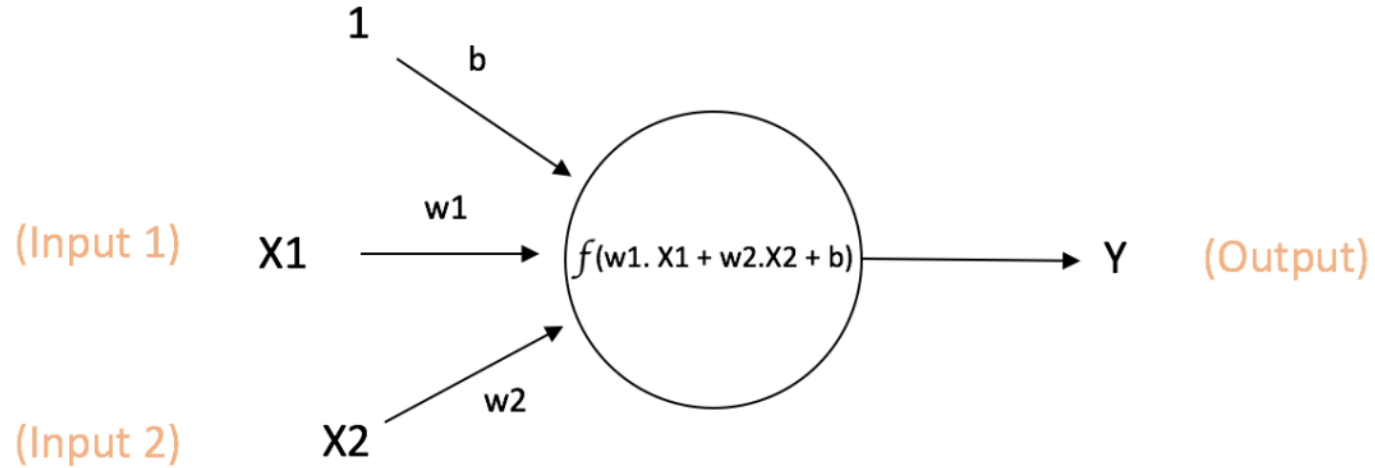


ReLU

$$\text{Output of neuron} = Y = f(w_1 \cdot X_1 + w_2 \cdot X_2 + b)$$

Image credit: <https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/>

A single neuron



$$\text{Output of neuron} = Y = f(w1 \cdot X1 + w2 \cdot X2 + b)$$

Composing neurons to represent more complex functions

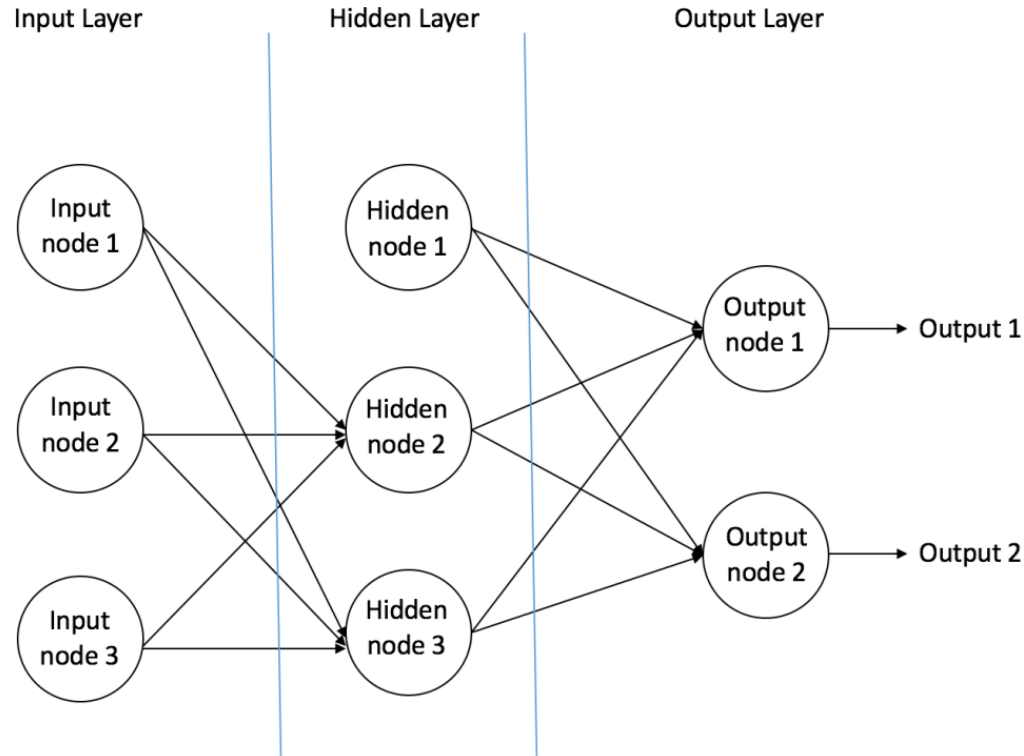
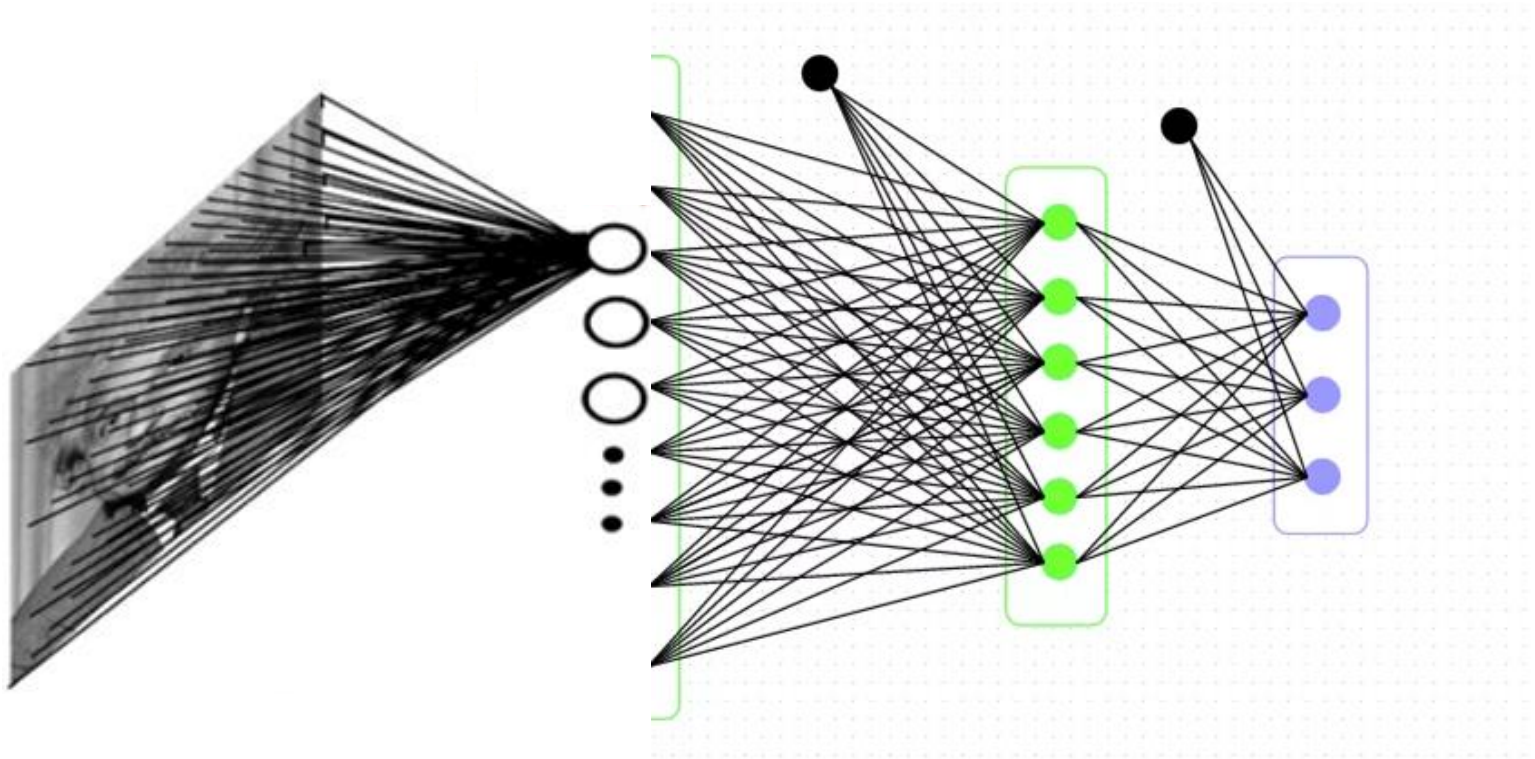
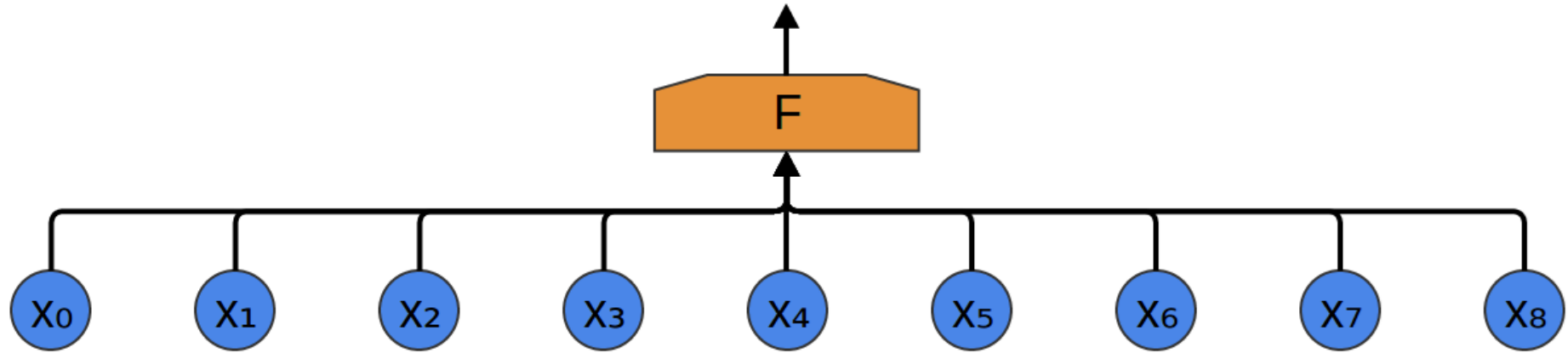


Image credit: <https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/>

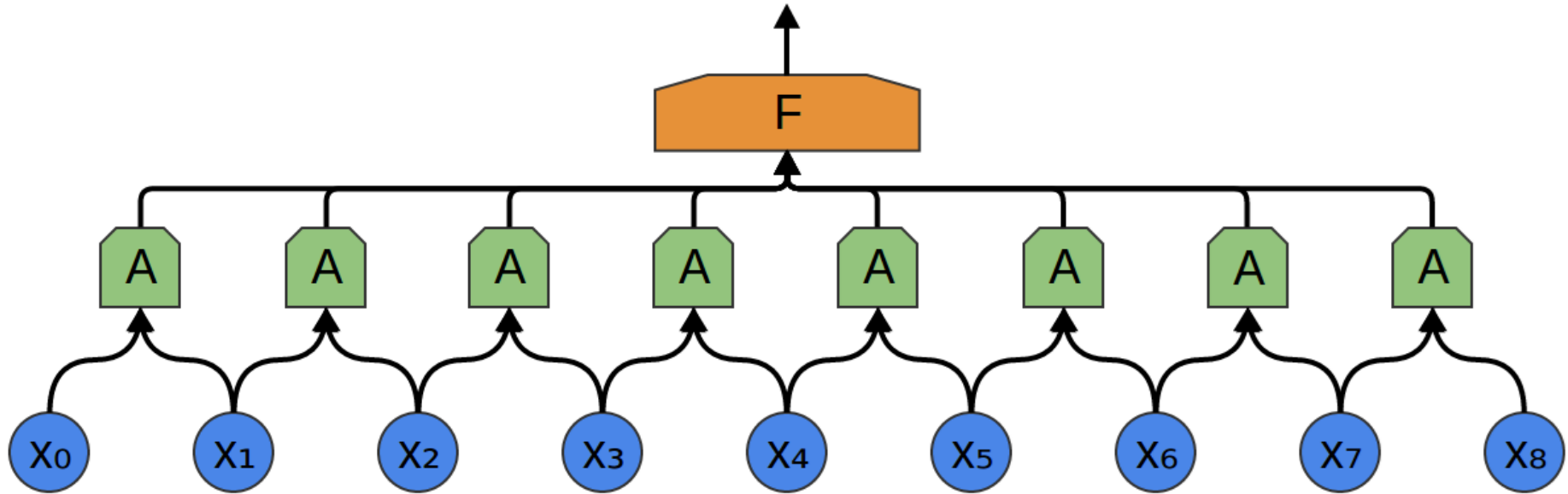
Fully connected network on the image



A fully connected layer



A convolutional layer



Repeat this across layers

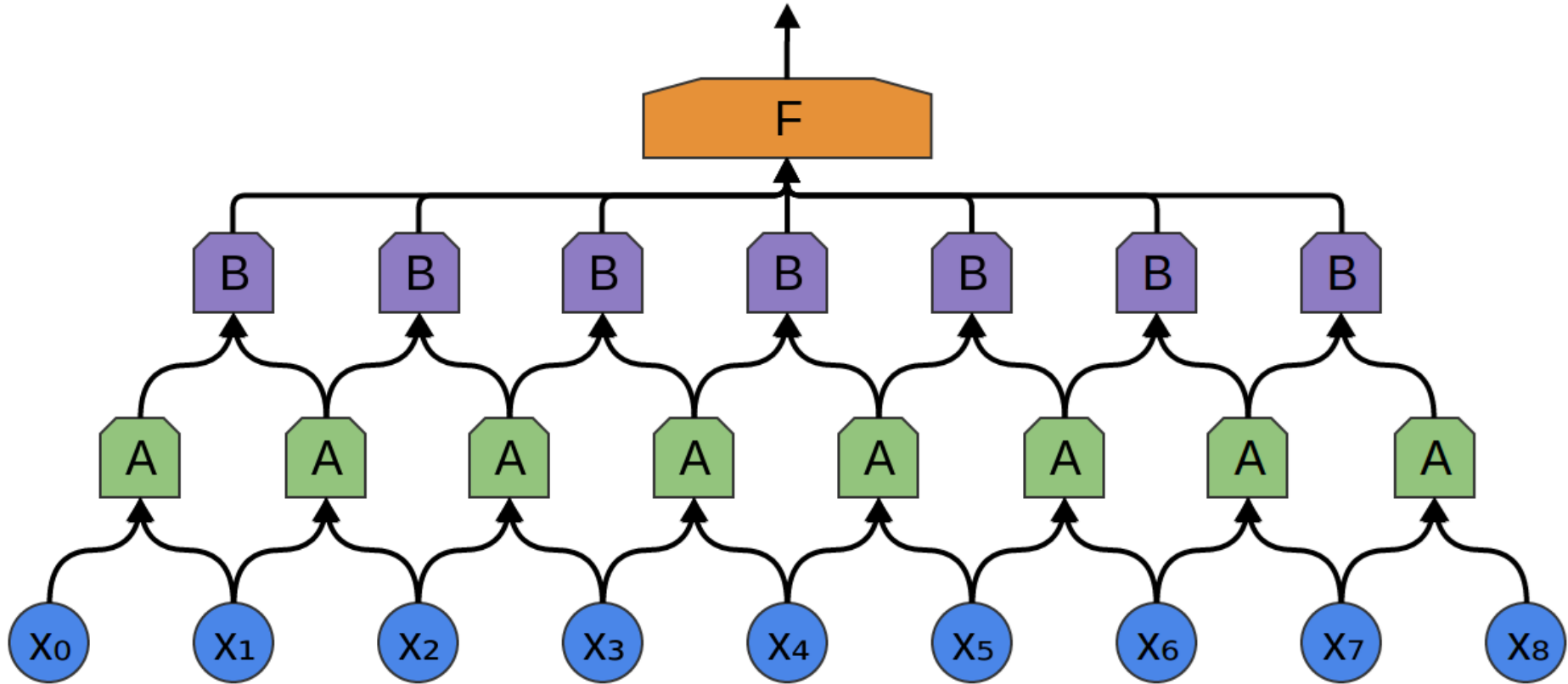


Image credit: <http://colah.github.io/posts/2014-07-Conv-Nets-Modular/>

Reduce dimensions by pooling (like a zoom out)

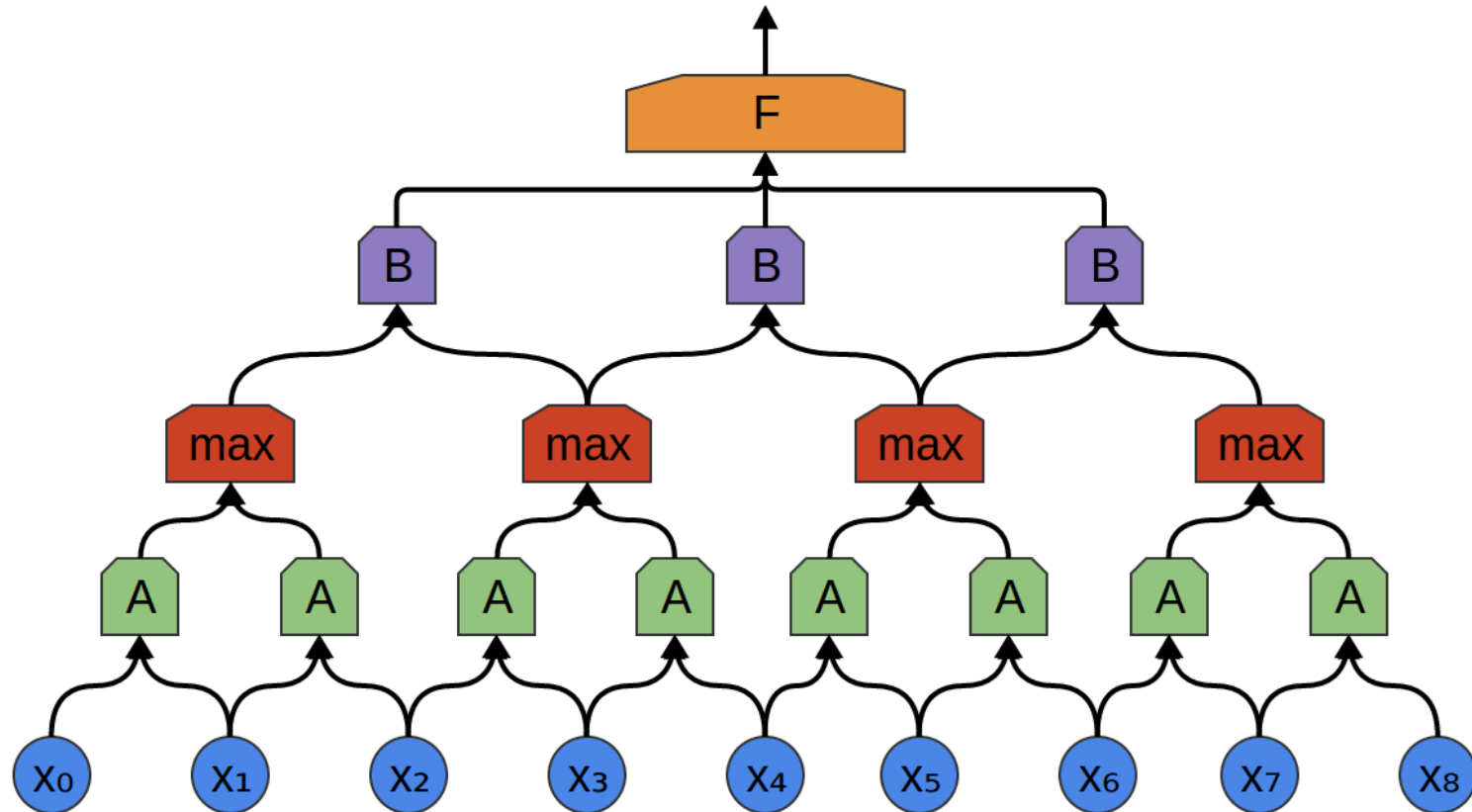


Image credit: <http://colah.github.io/posts/2014-07-Conv-Nets-Modular/>

2D version of the visualization

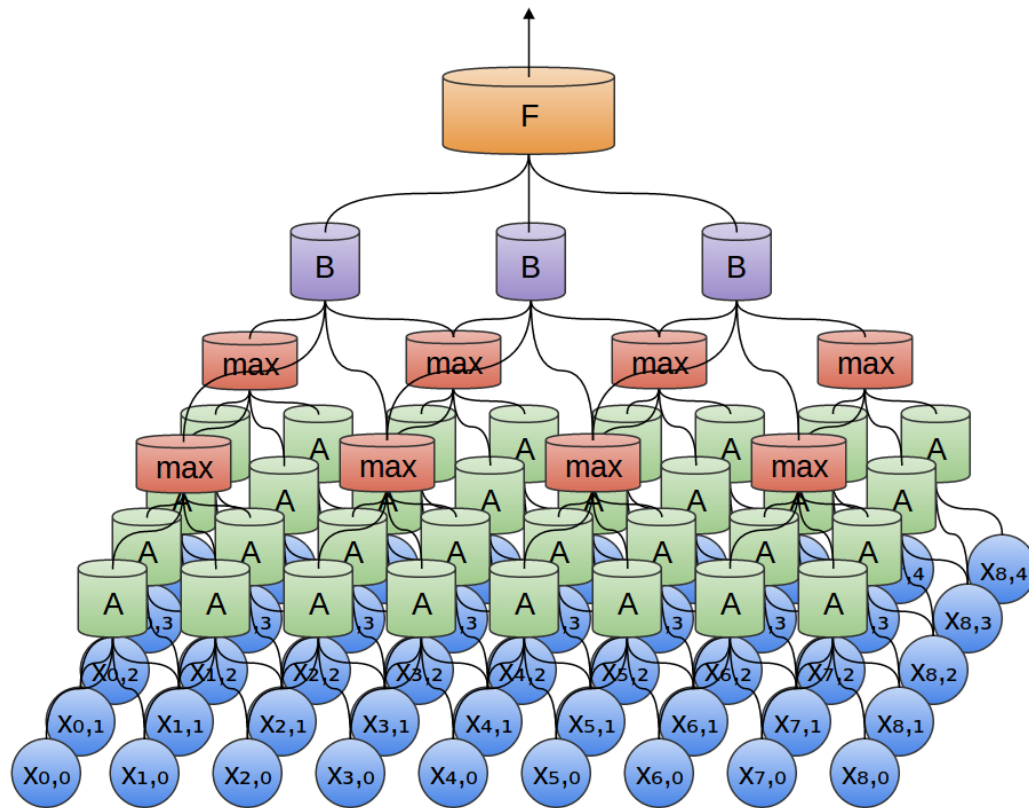
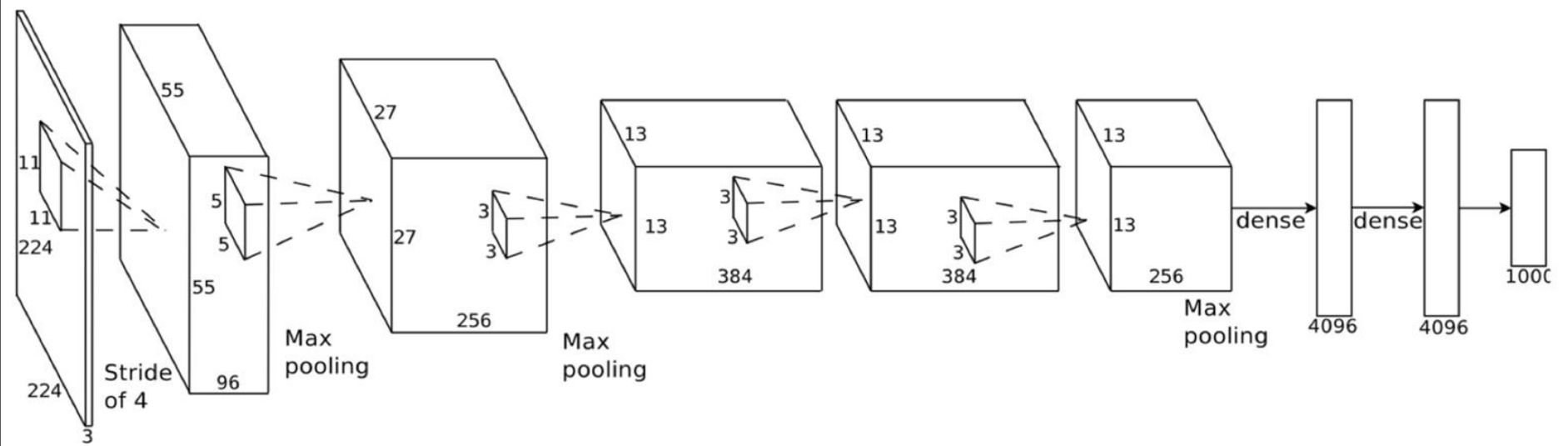
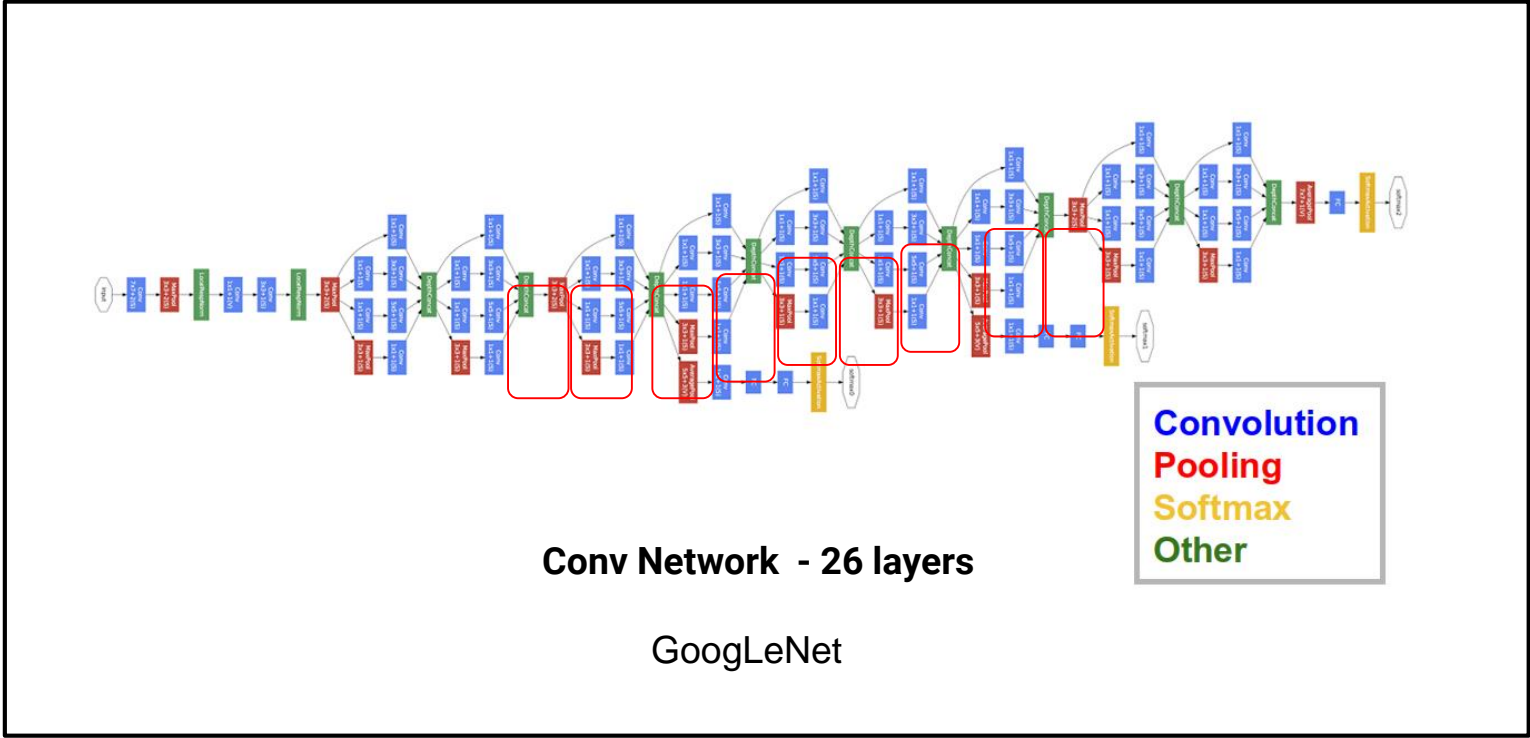


Image credit: <http://colah.github.io/posts/2014-07-Conv-Nets-Modular/>

A more abstract version of CNN

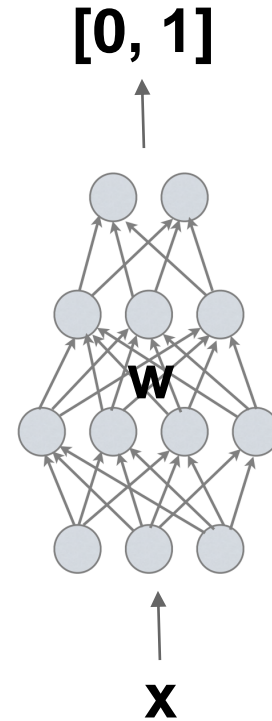


Even more abstract version of a CNN



Back to the original abstraction

$$f \left(\begin{array}{c} \text{[dotted grid]} \\ \mathbf{x} \end{array} ; \mathbf{w} \right) =$$



Loss functions for training the network

The parameters w are optimized to minimize a loss:

$$w_{\text{opt}} = \arg \min_w \sum_{i=1}^N L(f(x_i, w), y_i)$$

Index over training data set

Input representation of i^{th}
training data point

Desired output for
 i^{th} training point

Losses commonly used

Cross entropy loss (used in classification)

$$L(f(x_i, w), y_i) = -(y_i \log(f(x_i, w)) + (1 - y_i) \log(1 - f(x_i, w)))$$

These are either 0 or 1 (indicator variables denoting presence/absence of membership to a particular class)

L2 Loss (used in regression, when the predicted value is a continuous variable)

$$L(f(x_i, w), y_i) = \|y_i - f(x_i, w)\|^2$$

The optimization problem

$$L(w) = \min_w \sum_{i=1}^N L(f(x_i, w), y_i)$$

Use your favorite optimizer (Nesterov gradient, LBFGS, so on) to find the best w .

In practice N is very large, and x_i has very high dimensionality: These computational constraints restrict us to using Stochastic Gradient descent, which can be run in a distributed manner across several machines.

See: <https://research.google.com/pubs/pub40565.html>

Tensorflow abstracts out this complexity for you



<http://tensorflow.org/>

and

<https://github.com/tensorflow/tensorflow>

Open, standard software for
general machine learning

Great for Deep Learning in
particular

First released Nov 2015

Apache 2.0 license

Outline

- Deep dive into a specific Medical imaging problem (Diabetic retinopathy screening)
- Overview of Neural networks (specifically Convolutional neural network) and optimization.
- **The zoo of ML models and applications**

ML models and applications: Image classification

Steel drum



Ground truth

Steel drum
Folding chair
Loudspeaker

Accuracy: 1

Scale
T-shirt
Steel drum
Drumstick
Mud turtle

Accuracy: 1

Scale
T-shirt
Giant panda
Drumstick
Mud turtle

Accuracy: 0

ImageNet challenge: <https://arxiv.org/pdf/1409.0575.pdf>

Image classification models

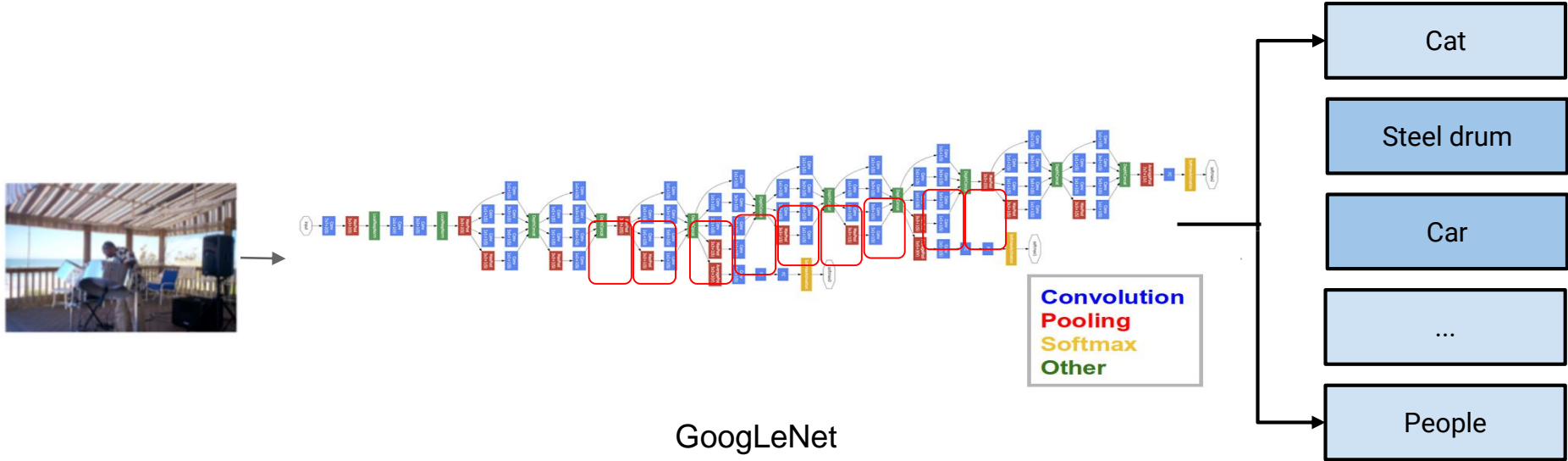
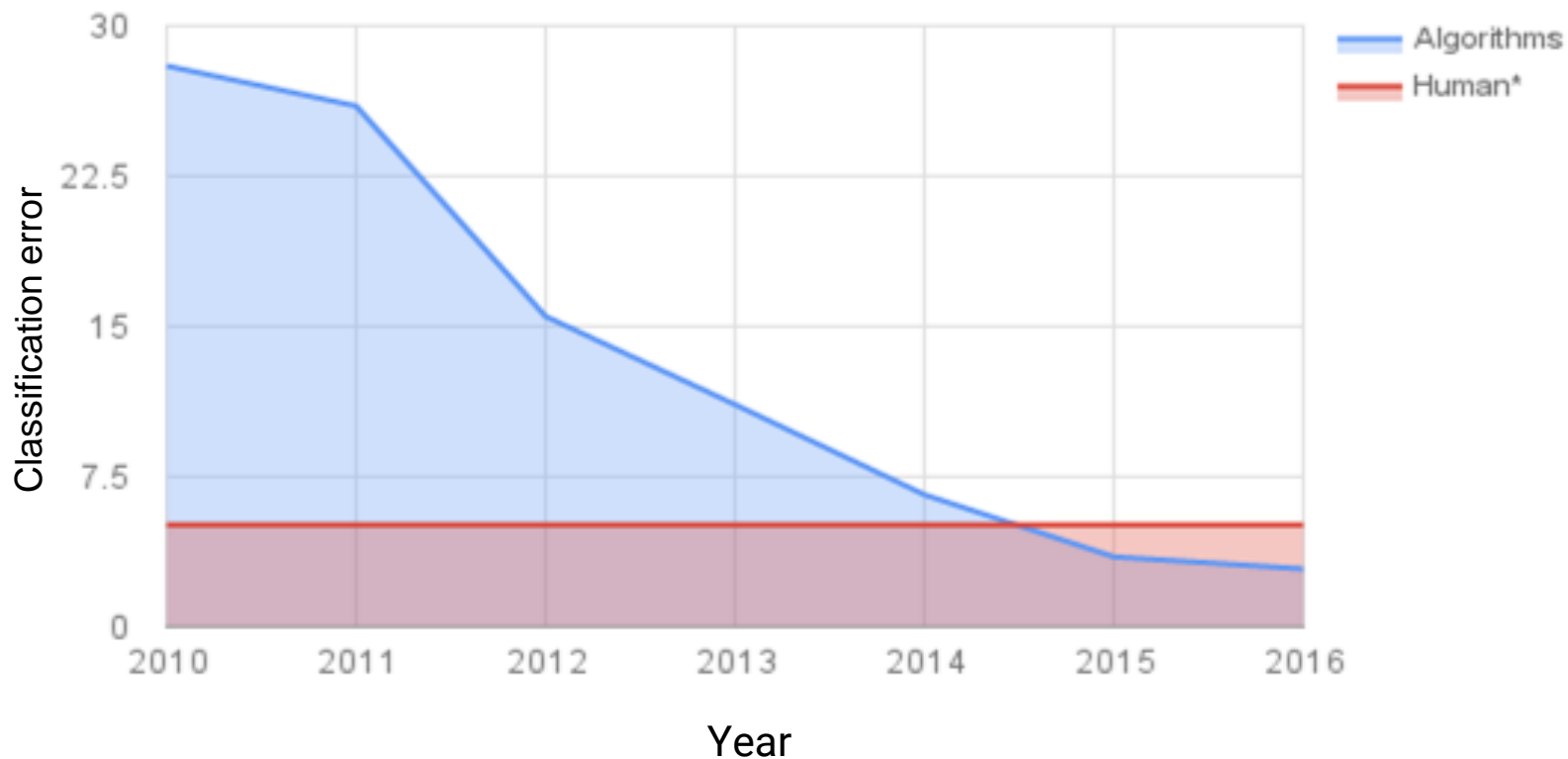
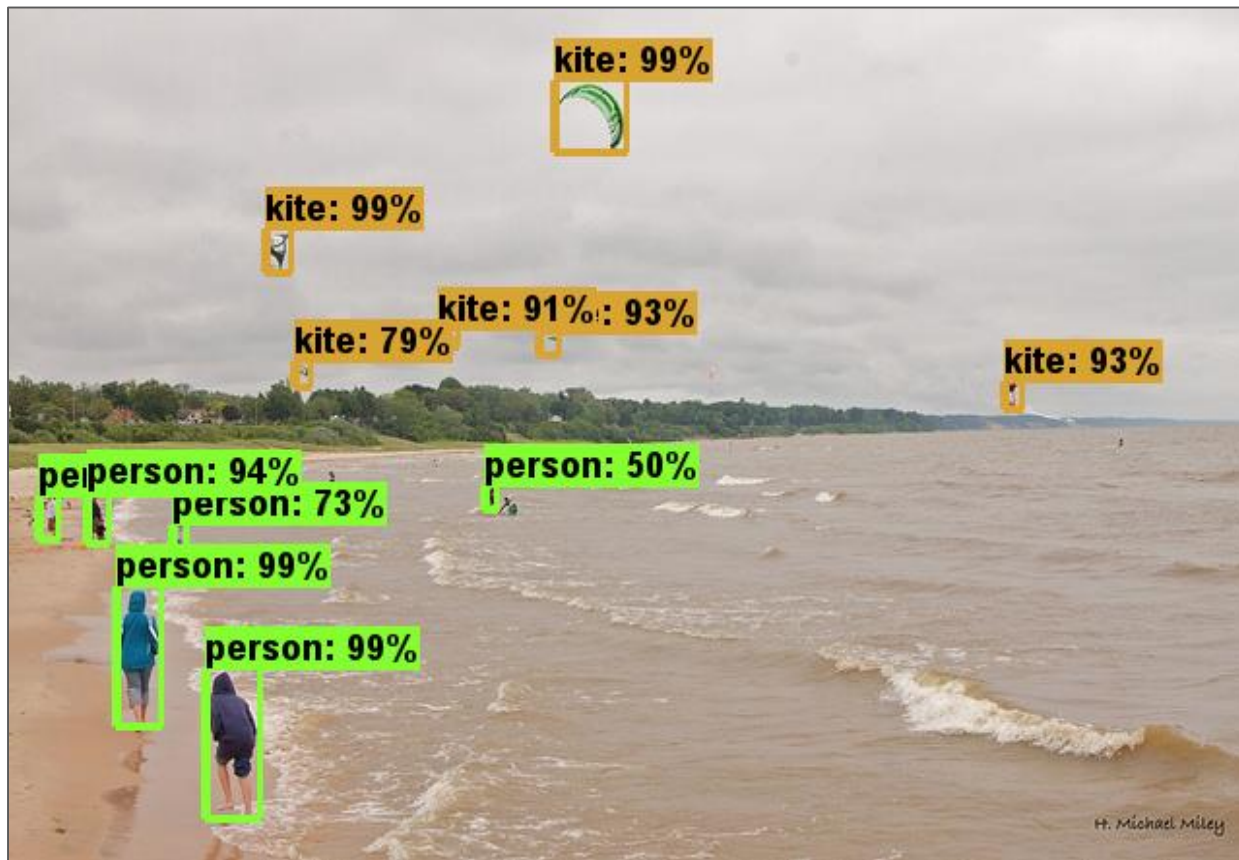


Image classification performance



* Human Performance based on analysis done by Andrej Karpathy. More details [here](#).

ML models and applications: Object detection



Object detection: Model architecture

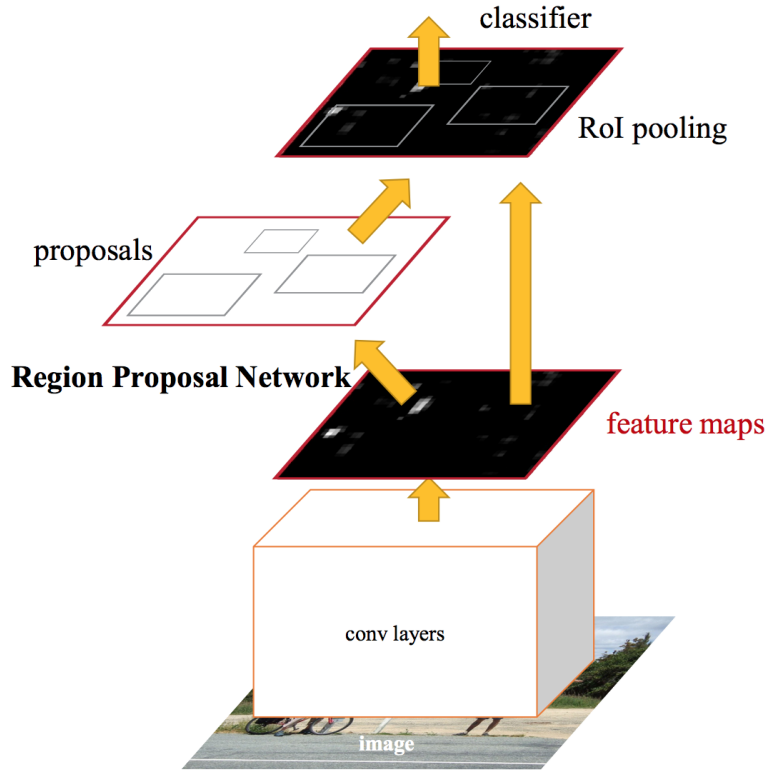


Image credit: <https://arxiv.org/pdf/1506.01497.pdf>

Object detection: Model performance

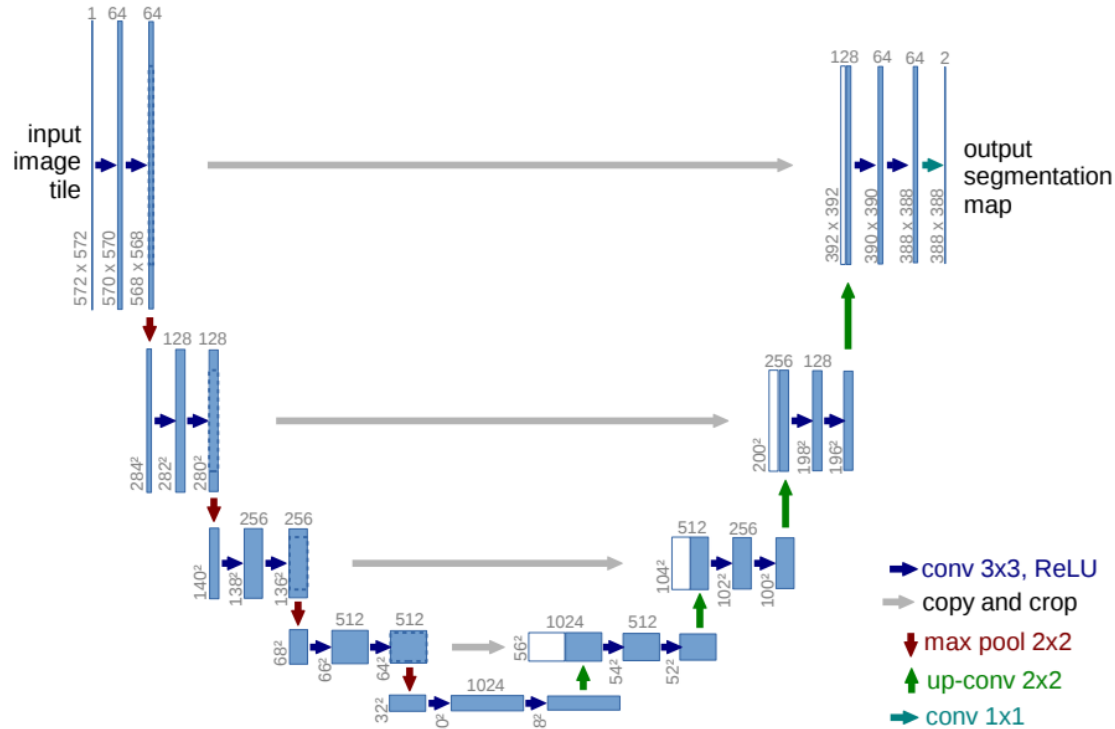
TODO: Get PR Curve and plot operating point

ML models and applications: Image segmentation



Image from: <http://cocodataset.org/#detections-challenge2017>

ML models and applications: Image segmentation



<https://arxiv.org/pdf/1505.04597.pdf>

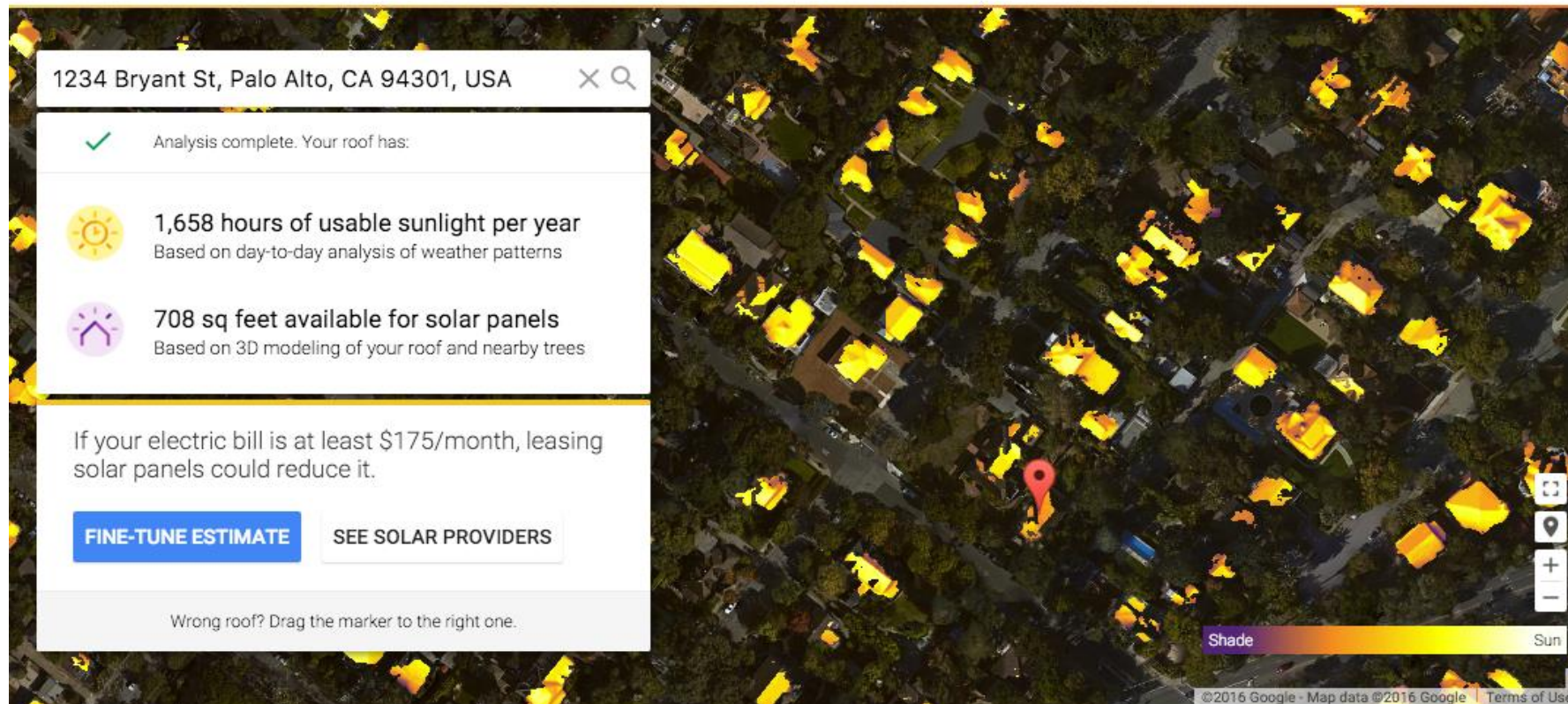
Image segmentation performance

Copy to Clipboard		Export to CSV			
		◀▶ mIoU ▼	fIoU ▶	mAcc ▶	pAcc ▶
+	ResNeXt-FPN	<u>0.287</u>	0.560	0.423	<u>0.694</u>
+	G-RMI	0.264	0.522	0.403	0.656
+	Oxford Active Vision Lab	0.232	0.505	0.339	0.653
+	Deeplab VGG-16	0.200	0.479	0.280	0.649
+	Vllab	0.124	0.394	0.176	0.581
Show 25 ▼ entries					

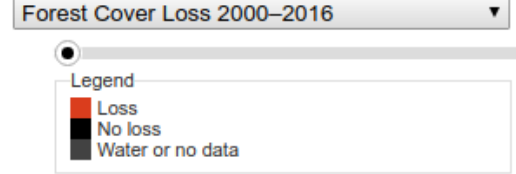
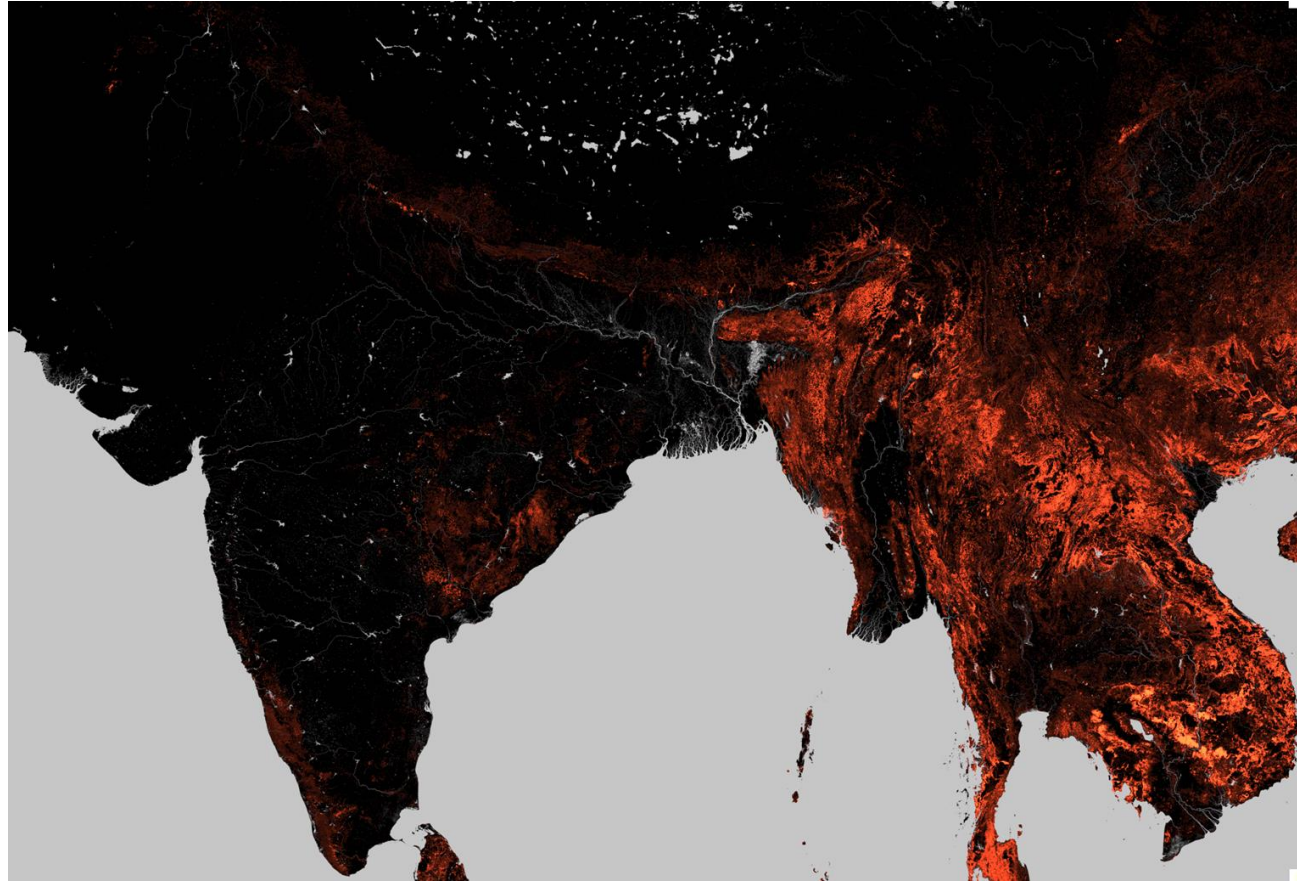
ML models and applications: Aerial image analysis

Google Project Sunroof

www.google.com/sunroof



ML models and applications: Satellite image analysis



Screenshot from live tool at:
<http://earthenginepartners.appspot.com/science-2013-global-forest>

ML models and applications: Translation

"Hello, how are you?" —————> "Bonjour, comment allez-vous?"

Translation model: LSTMs

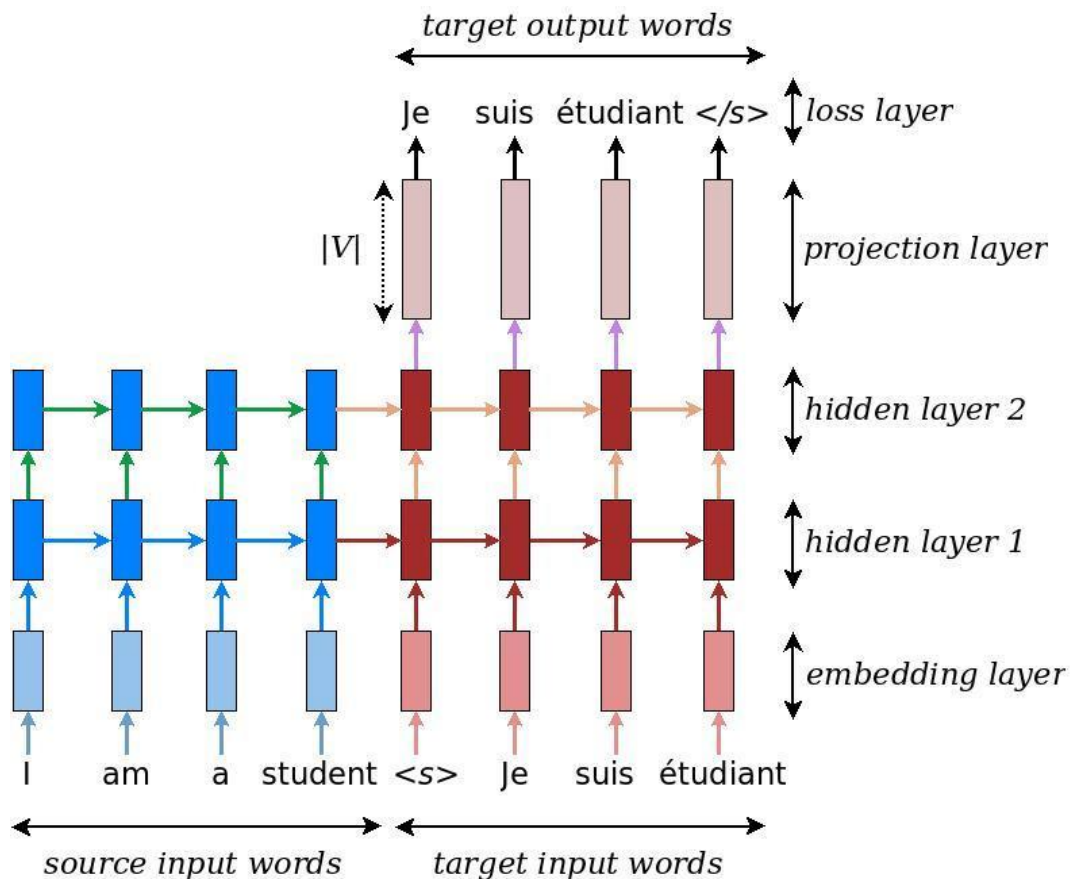


Image source:

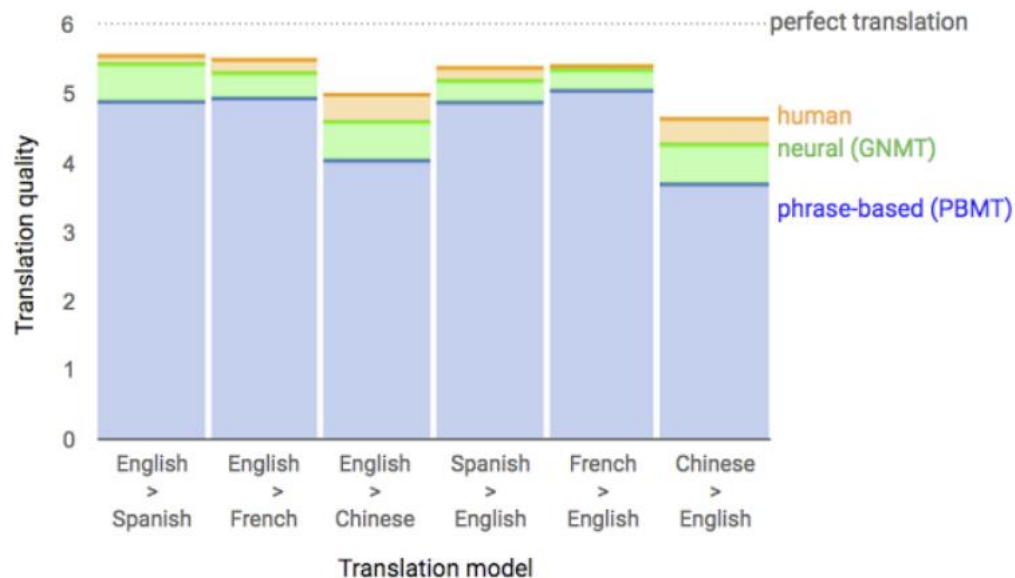
<https://www.tensorflow.org/tutorials/seq2seq>

Other Useful resources:

<http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/>

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Translation model performance



Data from side-by-side evaluations, where human raters compare the quality of translations for a given source sentence. Scores range from 0 to 6, with 0 meaning "completely nonsense translation", and 6 meaning "perfect translation."

Source: <https://research.googleblog.com/2016/09/a-neural-network-for-machine.html>

ML models and applications: Predicting molecule properties

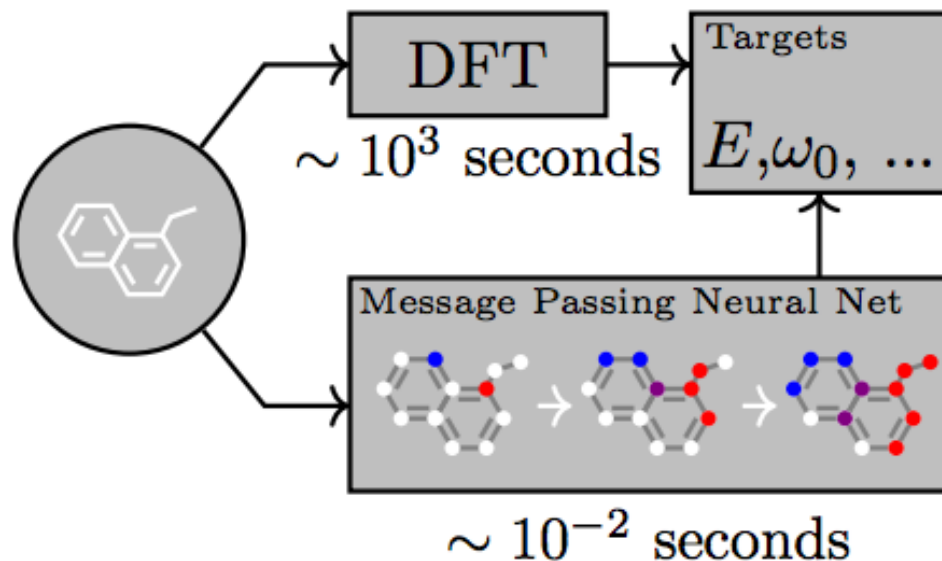


Figure 1. A Message Passing Neural Network predicts quantum properties of an organic molecule by modeling a computationally expensive DFT calculation.

<https://research.googleblog.com/2017/04/predicting-properties-of-molecules-with.html>

Predicting molecule properties: Performance

		μ	α	ϵ_{HOMO}	ϵ_{LUMO}	$\Delta\epsilon$	$\langle R^2 \rangle$	ZPVE	U_0	C_v	ω_1	NMMAE
		Debye	Bohr ³	eV	eV	eV	Bohr ²	eV	eV	cal/molK	cm ⁻¹	arb. u.
Mean		2.67	75.3	-6.54	0.322	6.86	1190	4.06	-76.6	31.6	3500	
MAD		1.17	6.29	0.439	1.05	1.07	203	0.717	8.19	3.21	238	
Target ^a		0.10	0.10	0.043	0.043	0.043	1.2	0.0012	0.043	0.050	10	
DFT ^b		0.09	0.74	2.0	2.6	1.19	-	0.0097	0.10	0.34	28	
EN	CM	0.844	1.33	0.338	0.631	0.722	55.5	0.0265	0.911	0.906	131	0.423
	BOB	0.763	1.20	0.283	0.521	0.614	55.3	0.0232	0.602	0.700	81.4	0.35
	BAML	0.686	0.793	0.186	0.275	0.339	32.6	0.0129	0.212	0.439	60.4	0.231
	ECFP4	0.737	3.45	0.224	0.344	0.383	118	0.270	3.68	1.51	86.6	0.462
	HDAD	0.563	0.437	0.139	0.238	0.278	6.19	0.00647	0.0983	0.0876	94.2	0.183
	HD	0.705	0.638	0.203	0.299	0.360	6.70	0.00949	0.192	0.195	104	0.236
	MARAD	0.707	0.698	0.222	0.305	0.391	27.4	0.00808	0.183	0.206	108	0.256
	Mean	0.715	1.22	0.228	0.373	0.441	43.1	0.0509	0.840	0.578	95.1	
BR	CM	0.844	1.33	0.338	0.632	0.723	55.5	0.0265	0.911	0.907	131	0.424
	BOB	0.761	1.14	0.279	0.521	0.614	48.0	0.0222	0.586	0.684	80.9	0.343
	BAML	0.685	0.785	0.183	0.275	0.339	30.4	0.0129	0.202	0.444	60.4	0.229
	ECFP4	0.737	3.45	0.224	0.344	0.383	118.0	0.270	3.69	1.51	86.7	0.462
	HDAD	0.565	0.43	0.14	0.238	0.278	5.94	0.00318	0.0614	0.0787	94.8	0.182
	HD	0.705	0.633	0.203	0.298	0.359	6.8	0.00693	0.171	0.19	104	0.235
	MARAD	0.647	0.533	0.18	0.257	0.315	26.8	0.00854	0.171	0.201	103	0.226
	Mean	0.706	1.19	0.221	0.367	0.430	41.7	0.0500	0.828	0.574	94.5	
RF	CM	0.608	1.04	0.208	0.302	0.373	45.0	0.0199	0.431	0.777	13.2	0.239
	BOB	0.450	0.623	0.120	0.137	0.164	39.0	0.0111	0.202	0.443	3.55	0.142
	BAML	0.434	0.638	0.107	0.118	0.141	51.1	0.0132	0.200	0.451	2.71	0.141
	ECFP4	0.483	3.70	0.143	0.145	0.166	109	0.242	3.66	1.57	14.7	0.349
	HDAD	0.454	1.71	0.116	0.136	0.156	48.3	0.0525	1.44	0.895	3.45	0.198
	HD	0.457	1.66	0.126	0.139	0.150	46.8	0.0497	1.39	0.879	4.18	0.197
	MARAD	0.607	0.676	0.178	0.243	0.311	45.3	0.0102	0.21	0.311	19.4	0.199
	Mean	0.499	1.43	0.142	0.174	0.209	54.9	0.0569	1.08	0.761	8.74	
KRR	CM	0.449	0.433	0.133	0.183	0.229	3.39	0.0048	0.128	0.118	33.5	0.136
	BOB	0.423	0.298	0.0948	0.122	0.148	0.978	0.00364	0.0667	0.0917	13.2	0.0981
	BAML	0.460	0.301	0.0946	0.121	0.152	3.9	0.00331	0.0519	0.082	19.9	0.105
	ECFP4	0.490	4.17	0.124	0.133	0.174	128	0.248	4.25	1.84	26.7	0.383
	HDAD	0.334	0.175	0.0662	0.0842	0.107	1.62	0.00191	0.0251	0.0441	23.1	0.0768
	HD	0.364	0.299	0.0874	0.113	0.143	1.72	0.00316	0.0644	0.0844	21.3	0.0935
	MARAD	0.468	0.343	0.103	0.124	0.163	7.58	0.00301	0.0529	0.0758	21.3	0.112
	Mean	0.427	0.859	0.101	0.126	0.159	21.1	0.0383	0.662	0.333	22.7	
GG	MG	0.238	0.151	0.0587	0.0564	0.0835	5.98	0.00291	0.0317	0.0724	6.32	0.058
GC	MG	0.0696	0.227	0.0509	0.0471	0.0766	5.68	0.00975	0.13	0.0892	3.15	0.0427

<https://arxiv.org/abs/1702.05532>

Predicting molecule properties: Performance

		μ	α	ϵ_{HOMO}	ϵ_{LUMO}	$\Delta\epsilon$	$\langle R^2 \rangle$	ZPVE	U_0	C_v	ω_1	NMMAE
		Debye	Bohr ³	eV	eV	eV	Bohr ²	eV	eV	cal/molK	cm ⁻¹	arb. u.
Mean		2.67	75.3	-6.54	0.322	6.86	1190	4.06	-76.6	31.6	3500	
MAD		1.17	6.29	0.439	1.05	1.07	203	0.717	8.19	3.21	238	
Target ^a		0.10	0.10	0.043	0.043	0.043	1.2	0.0012	0.043	0.050	10	
DFT ^b		0.09	0.74	2.0	2.6	1.19	-	0.0097	0.10	0.34	28	

Various molecular properties

EN	CM	0.844	1.33	0.338	0.631	0.722	55.5	0.0265	0.911	0.906	131	0.423
	BOB	0.763	1.20	0.283	0.521	0.614	55.3	0.0232	0.602	0.700	81.4	0.35
	BAML	0.686	0.793	0.186	0.275	0.339	32.6	0.0129	0.212	0.439	60.4	0.231
	ECFP4	0.737	3.45	0.224	0.344	0.383	118	0.270	3.68	1.51	86.6	0.462
	HDAD	0.563	0.437	0.139	0.238	0.278	6.19	0.00647	0.0983	0.0876	94.2	0.183
	HD	0.705	0.638	0.203	0.299	0.360	6.70	0.00949	0.192	0.195	104	0.236
	MARAD	0.707	0.698	0.222	0.305	0.391	27.4	0.00808	0.183	0.206	108	0.256
	Mean	0.715	1.22	0.228	0.373	0.441	43.1	0.0509	0.840	0.578	95.1	
BR	CM	0.844	1.33	0.338	0.632	0.723	55.5	0.0265	0.911	0.907	131	0.424
	BOB	0.761	1.14	0.279	0.521	0.614	48.0	0.0222	0.586	0.684	80.9	0.343
	BAML	0.685	0.785	0.183	0.275	0.339	30.4	0.0129	0.202	0.444	60.4	0.229
	ECFP4	0.737	3.45	0.224	0.344	0.383	118.0	0.270	3.69	1.51	86.7	0.462
	HDAD	0.565	0.43	0.14	0.238	0.278	5.94	0.00318	0.0614	0.0787	94.8	0.182
	HD	0.705	0.633	0.203	0.298	0.359	6.8	0.00693	0.171	0.19	104	0.235
	MARAD	0.647	0.533	0.18	0.257	0.315	26.8	0.00854	0.171	0.201	103	0.226
	Mean	0.706	1.19	0.221	0.367	0.430	41.7	0.0500	0.828	0.574	94.5	
RF	CM	0.608	1.04	0.208	0.302	0.373	45.0	0.0199	0.431	0.777	13.2	0.239
	BOB	0.450	0.623	0.120	0.137	0.164	39.0	0.0111	0.202	0.443	3.55	0.142
	BAML	0.434	0.638	0.107	0.118	0.141	51.1	0.0132	0.200	0.451	2.71	0.141
	ECFP4	0.483	3.70	0.143	0.145	0.166	109	0.242	3.66	1.57	14.7	0.349
	HDAD	0.454	1.71	0.116	0.136	0.156	48.3	0.0525	1.44	0.895	3.45	0.198
	HD	0.457	1.66	0.126	0.139	0.150	46.8	0.0497	1.39	0.879	4.18	0.197
	MARAD	0.607	0.676	0.178	0.243	0.311	45.3	0.0102	0.21	0.311	19.4	0.199
	Mean	0.499	1.43	0.142	0.174	0.209	54.9	0.0569	1.08	0.761	8.74	
KRR	CM	0.449	0.433	0.133	0.183	0.229	3.39	0.0048	0.128	0.118	33.5	0.136
	BOB	0.423	0.298	0.0948	0.122	0.148	0.978	0.00364	0.0667	0.0917	13.2	0.0981
	BAML	0.460	0.301	0.0946	0.121	0.152	3.9	0.00331	0.0519	0.082	19.9	0.105
	ECFP4	0.490	4.17	0.124	0.133	0.174	128	0.248	4.25	1.84	26.7	0.383
	HDAD	0.334	0.175	0.0662	0.0842	0.107	1.62	0.00191	0.0251	0.0441	23.1	0.0768
	HD	0.364	0.299	0.0874	0.113	0.143	1.72	0.00316	0.0644	0.0844	21.3	0.0935
	MARAD	0.468	0.343	0.103	0.124	0.163	7.58	0.00301	0.0529	0.0758	21.3	0.112
	Mean	0.427	0.859	0.101	0.126	0.159	21.1	0.0383	0.662	0.333	22.7	
GG	MG	0.238	0.151	0.0587	0.0564	0.0835	5.98	0.00291	0.0317	0.0724	6.32	0.058
GC	MG	0.0696	0.227	0.0509	0.0471	0.0766	5.68	0.00975	0.13	0.0892	3.15	0.0427

Different ML methods

<https://arxiv.org/abs/1702.05532>

Predicting molecule properties: Performance

		μ	α	ϵ_{HOMO}	ϵ_{LUMO}	$\Delta\epsilon$	$\langle R^2 \rangle$	ZPVE	U_0	C_v	ω_1	NMMAE
		Debye	Bohr ³	eV	eV	eV	Bohr ²	eV	eV	cal/molK	cm ⁻¹	arb. u.
Mean		2.67	75.3	-6.54	0.322	6.86	1190	4.06	-76.6	31.6	3500	
MAD		1.17	6.29	0.439	1.05	1.07	203	0.717	8.19	3.21	238	
Target ^a		0.10	0.10	0.043	0.043	0.043	1.2	0.0012	0.043	0.050	10	
DFT ^b		0.09	0.74	2.0	2.6	1.19	-	0.0097	0.10	0.34	28	

Various molecular properties

EN	CM	0.844	1.33	0.338	0.631	0.722	55.5	0.0265	0.911	0.906	131	0.423
	BOB	0.763	1.20	0.283	0.521	0.614	55.3	0.0232	0.602	0.700	81.4	0.35
	BAML	0.686	0.793	0.186	0.275	0.339	32.6	0.0129	0.212	0.439	60.4	0.231
	ECFP4	0.737	3.45	0.224	0.344	0.383	118	0.270	3.68	1.51	86.6	0.462
	HDAD	0.563	0.437	0.139	0.238	0.278	6.19	0.00647	0.0983	0.0876	94.2	0.183
	HD	0.705	0.638	0.203	0.299	0.360	6.70	0.00949	0.192	0.195	104	0.236
	MARAD	0.707	0.698	0.222	0.305	0.391	27.4	0.00808	0.183	0.206	108	0.256
	Mean	0.715	1.22	0.228	0.373	0.441	43.1	0.0509	0.840	0.578	95.1	

Different ML methods

BR	CM	0.844	1.33	0.338	0.632	0.723	55.5	0.0265	0.911	0.907	131	0.424
	BOB	0.761	1.14	0.279	0.521	0.614	48.0	0.0222	0.586	0.684	80.9	0.343
	BAML	0.685	0.785	0.183	0.275	0.339	30.4	0.0129	0.202	0.444	60.4	0.229
	ECFP4	0.737	3.45	0.224	0.344	0.383	118.0	0.270	3.69	1.51	86.7	0.462
	HDAD	0.565	0.43	0.14	0.238	0.278	5.94	0.00318	0.0614	0.0787	94.8	0.182
	HD	0.705	0.633	0.203	0.298	0.359	6.8	0.00693	0.171	0.19	104	0.235
	MARAD	0.647	0.533	0.18	0.257	0.315	26.8	0.00854	0.171	0.201	103	0.226
	Mean	0.706	1.19	0.221	0.367	0.430	41.7	0.0500	0.828	0.574	94.5	

<https://arxiv.org/abs/1702.05532>

RF	CM	0.608	1.04	0.208	0.302	0.373	45.0	0.0199	0.431	0.777	13.2	0.239
	BOB	0.450	0.623	0.120	0.137	0.164	39.0	0.0111	0.202	0.443	3.55	0.142
	BAML	0.434	0.638	0.107	0.118	0.141	51.1	0.0132	0.200	0.451	2.71	0.141
	ECFP4	0.483	3.70	0.143	0.145	0.166	109	0.242	3.66	1.57	14.7	0.349
	HDAD	0.454	1.71	0.116	0.136	0.156	48.3	0.0525	1.44	0.895	3.45	0.198
	HD	0.457	1.66	0.126	0.139	0.150	46.8	0.0497	1.39	0.879	4.18	0.197
	MARAD	0.607	0.676	0.178	0.243	0.311	45.3	0.0102	0.21	0.311	19.4	0.199
	Mean	0.499	1.43	0.142	0.174	0.209	54.9	0.0569	1.08	0.761	8.74	

KRR	CM	0.449	0.433	0.133	0.183	0.229	3.39	0.0048	0.128	0.118	33.5	0.136
	BOB	0.423	0.298	0.0948	0.122	0.148	0.978	0.00364	0.0667	0.0917	13.2	0.0981
	BAML	0.460	0.301	0.0946	0.121	0.152	3.9	0.00331	0.0519	0.082	19.9	0.105
	ECFP4	0.490	4.17	0.124	0.133	0.174	128	0.248	4.25	1.84	26.7	0.383
	HDAD	0.334	0.175	0.0662	0.0842	0.107	1.62	0.00191	0.0251	0.0441	23.1	0.0768
	HD	0.364	0.299	0.0874	0.113	0.143	1.72	0.00316	0.0644	0.0844	21.3	0.0935
	MARAD	0.468	0.343	0.103	0.124	0.163	7.58	0.00301	0.0529	0.0758	21.3	0.112
	Mean	0.427	0.850	0.101	0.126	0.150	21.1	0.0082	0.662	0.323	22.7	

Neural network based ML methods

GG	MG	0.238	0.151	0.0587	0.0564	0.0835	5.98	0.00291	0.0317	0.0724	6.32	0.058
GC	MG	0.0696	0.227	0.0509	0.0471	0.0766	5.68	0.00975	0.13	0.0892	3.15	0.0427

A Monsoon prediction paper!

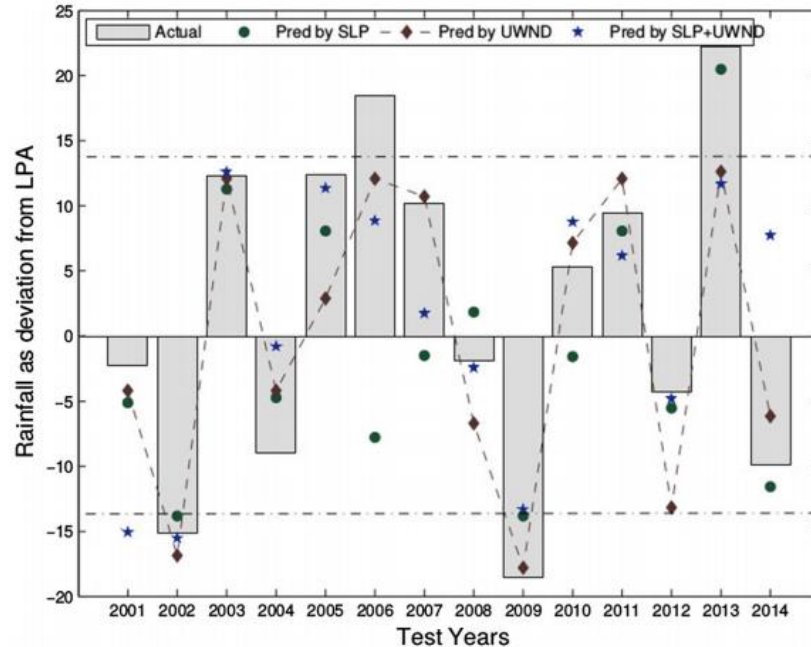


Figure 8. Forecast of the central India summer monsoon (June–September) by SLP, UWND and SLP+UWND during 2001–2014.