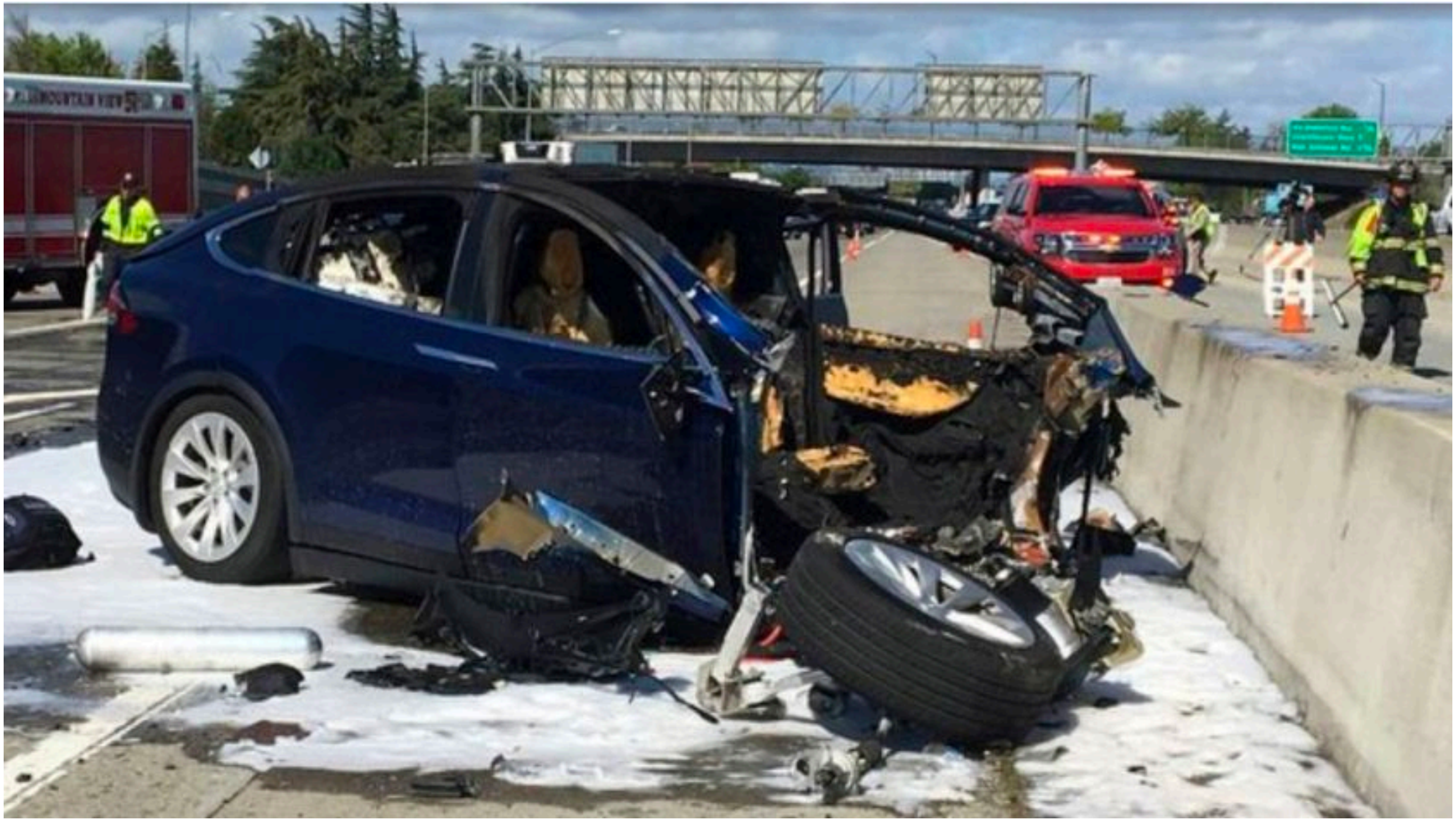
인공지능의 불확실성

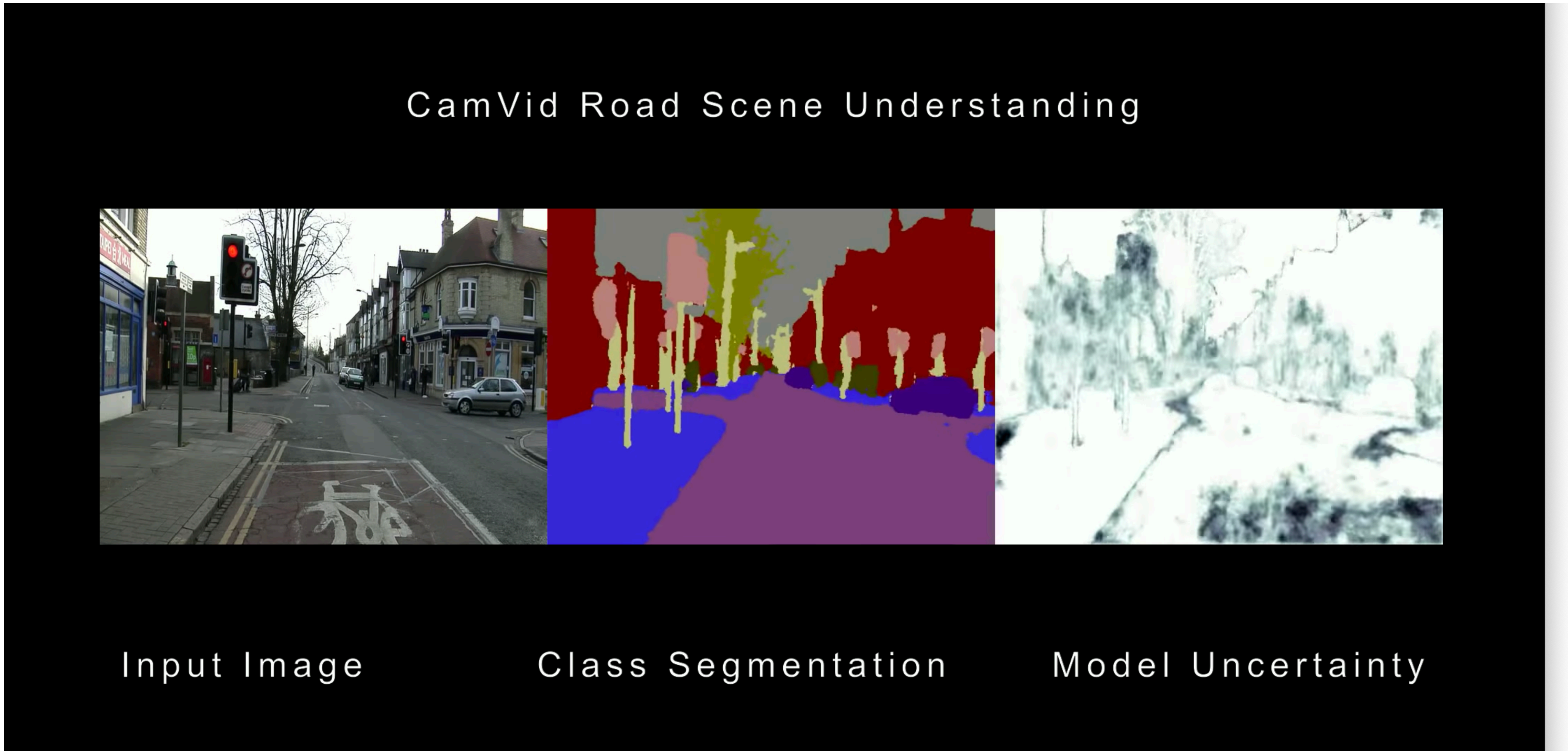
만약 목숨이 달린 일이라면 딥러닝에게 맡기시겠습니까?

https://youtu.be/0yCYro9H1kM

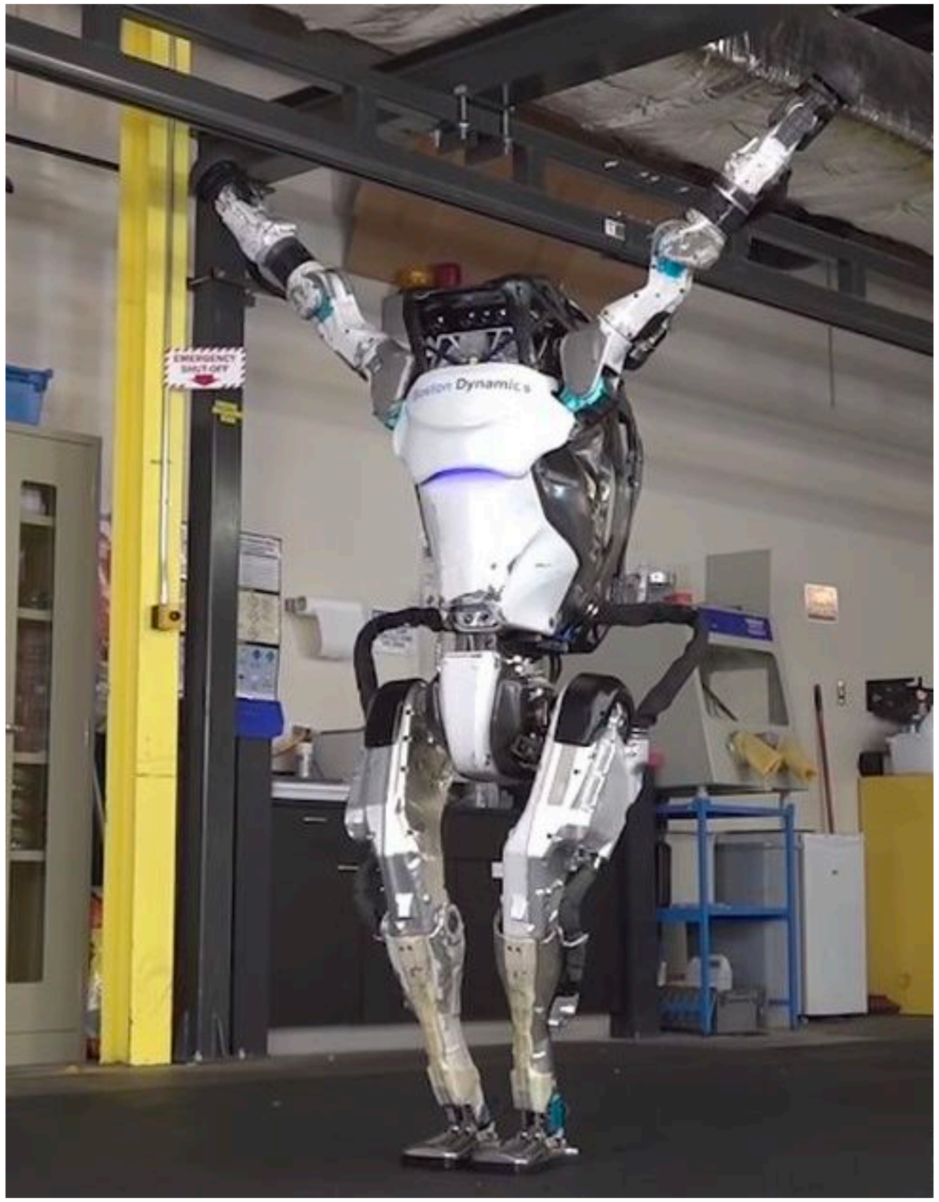
https://sophia.smith.edu/~jmoulton/Hawaii/pages/m%20truck%20reflect.htm



https://www.latimes.com/business/hiltzik/la-fi-hiltzik-tesla-autopilot-flaw-20190522-story.html

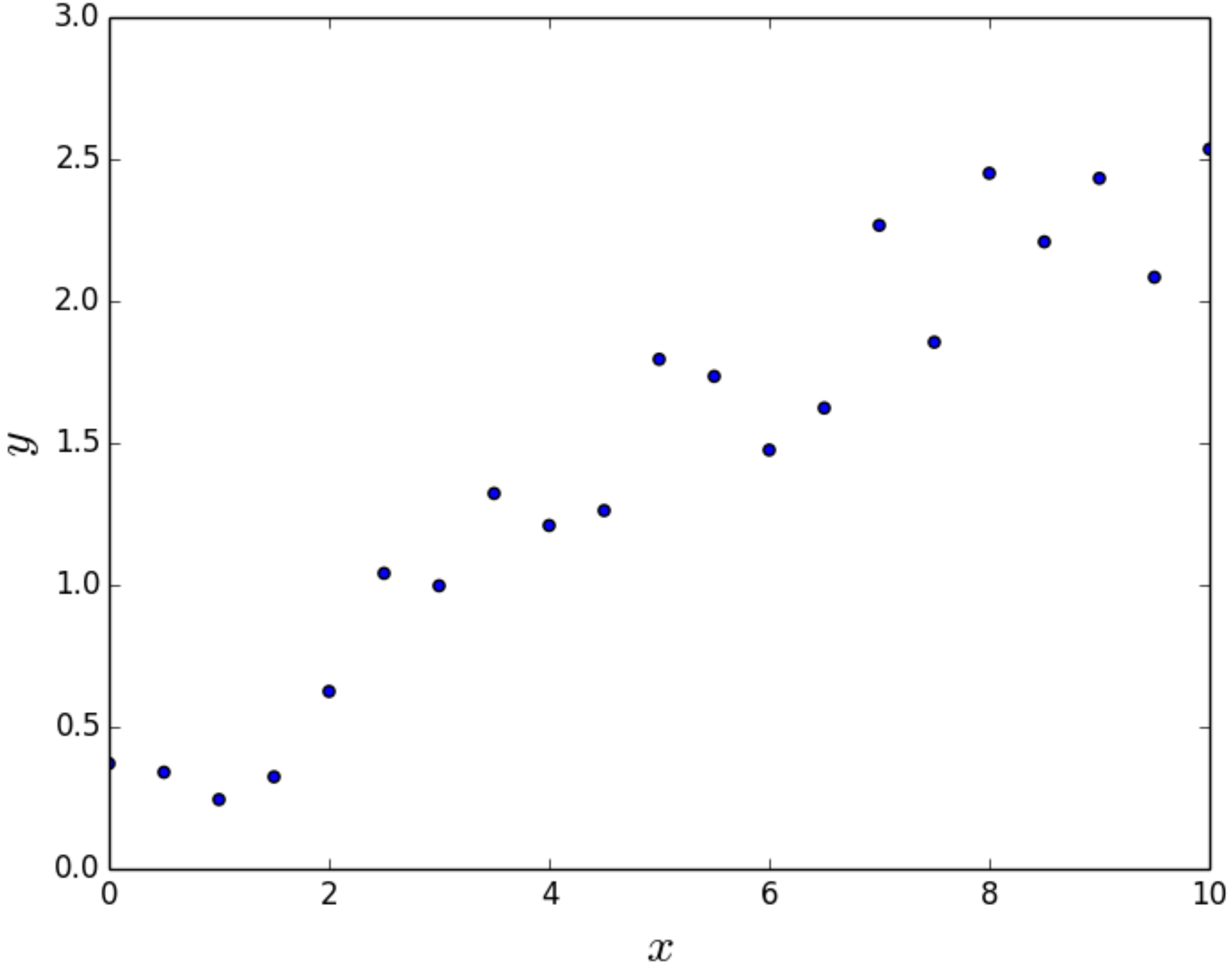


https://youtu.be/Zu1BzW8fAjw?t=7

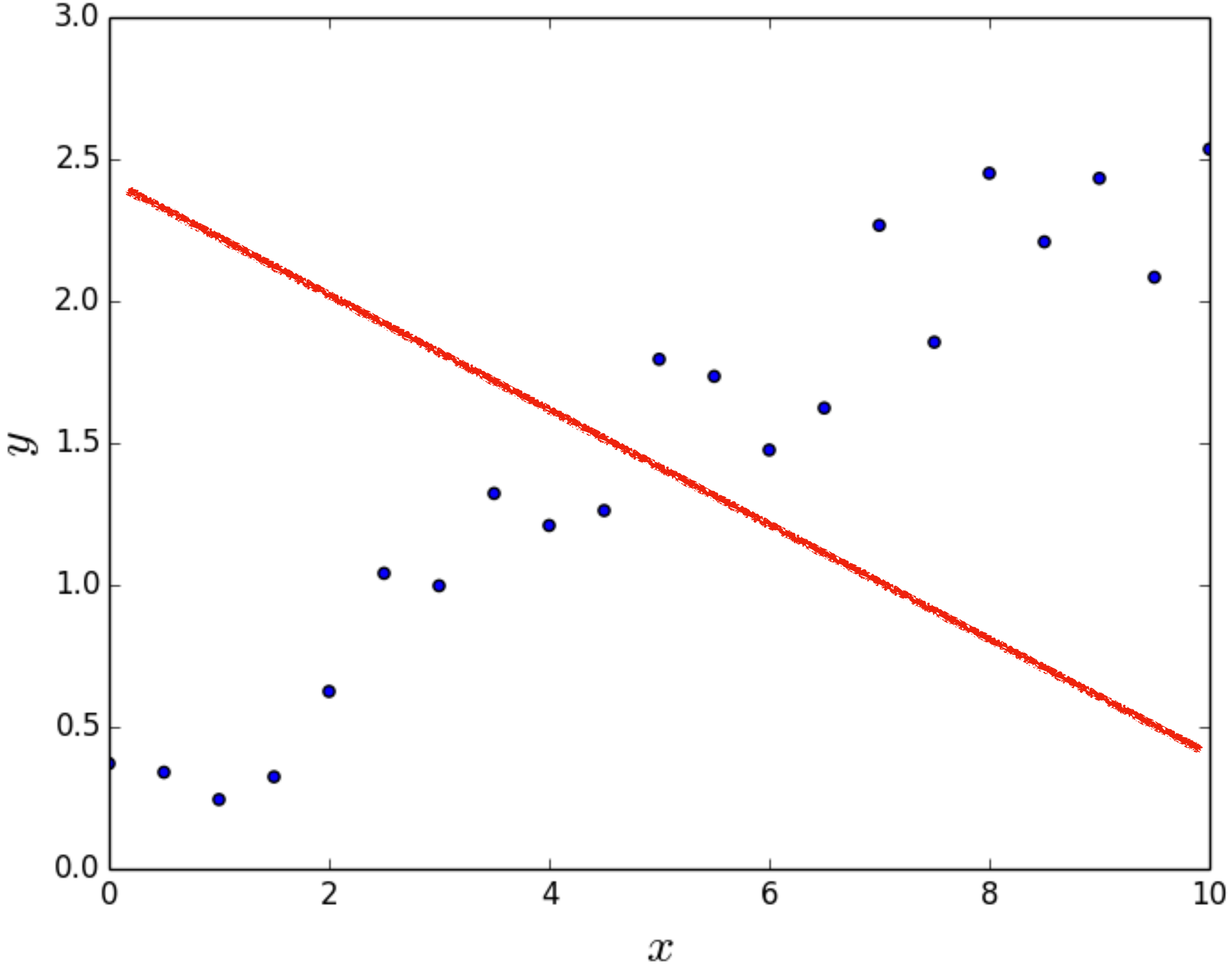


https://www.youtube.com/watch?v=_sBBaNYex3E

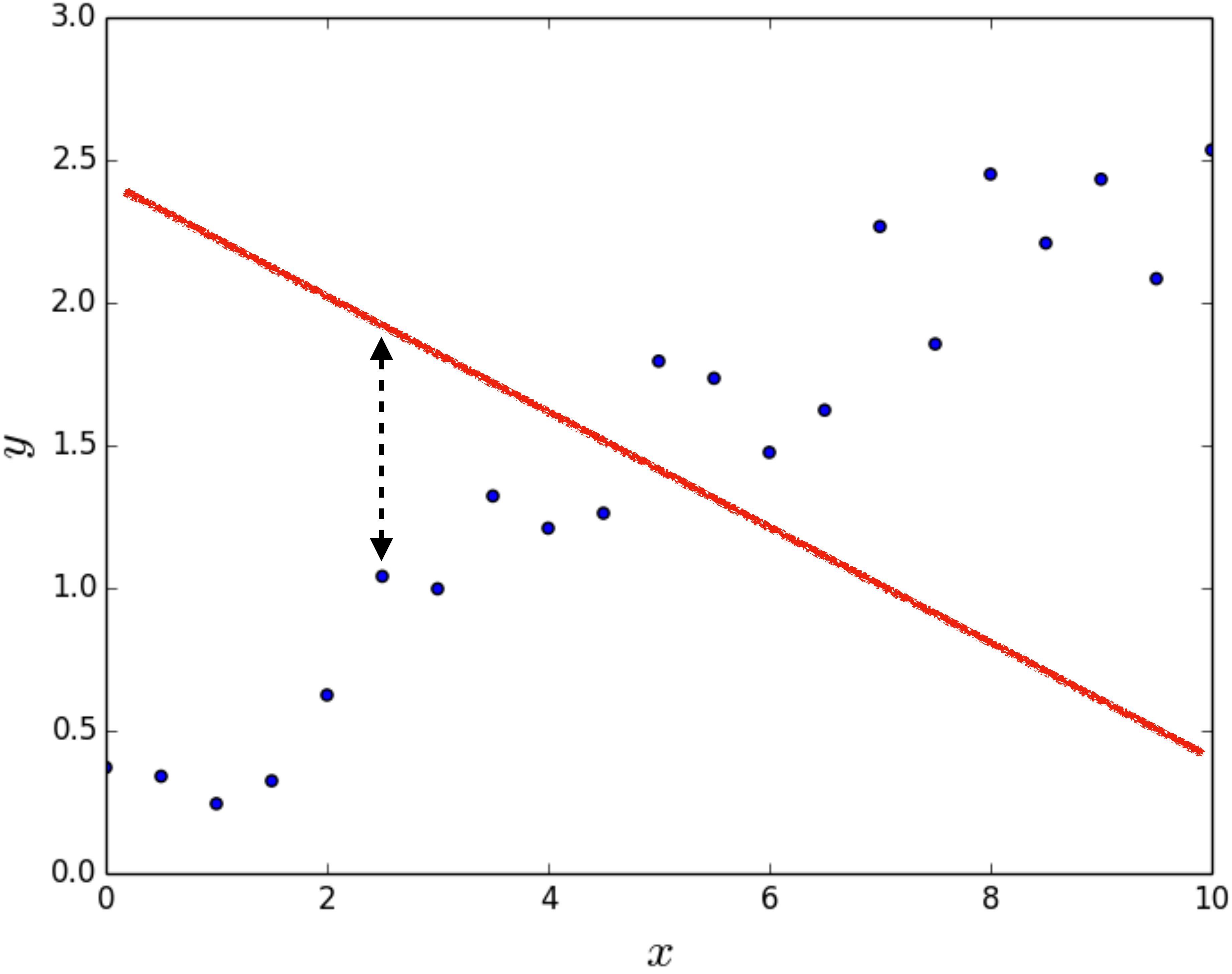
https://scienceblog.cancerresearchuk.org/2019/05/09/prostate-cancer-mri-scans-is-the-nhs-ready/


3.0
2.5
2.0
1.5
1.0
0.5
0.0
y
0
2
4
6
8
10
x

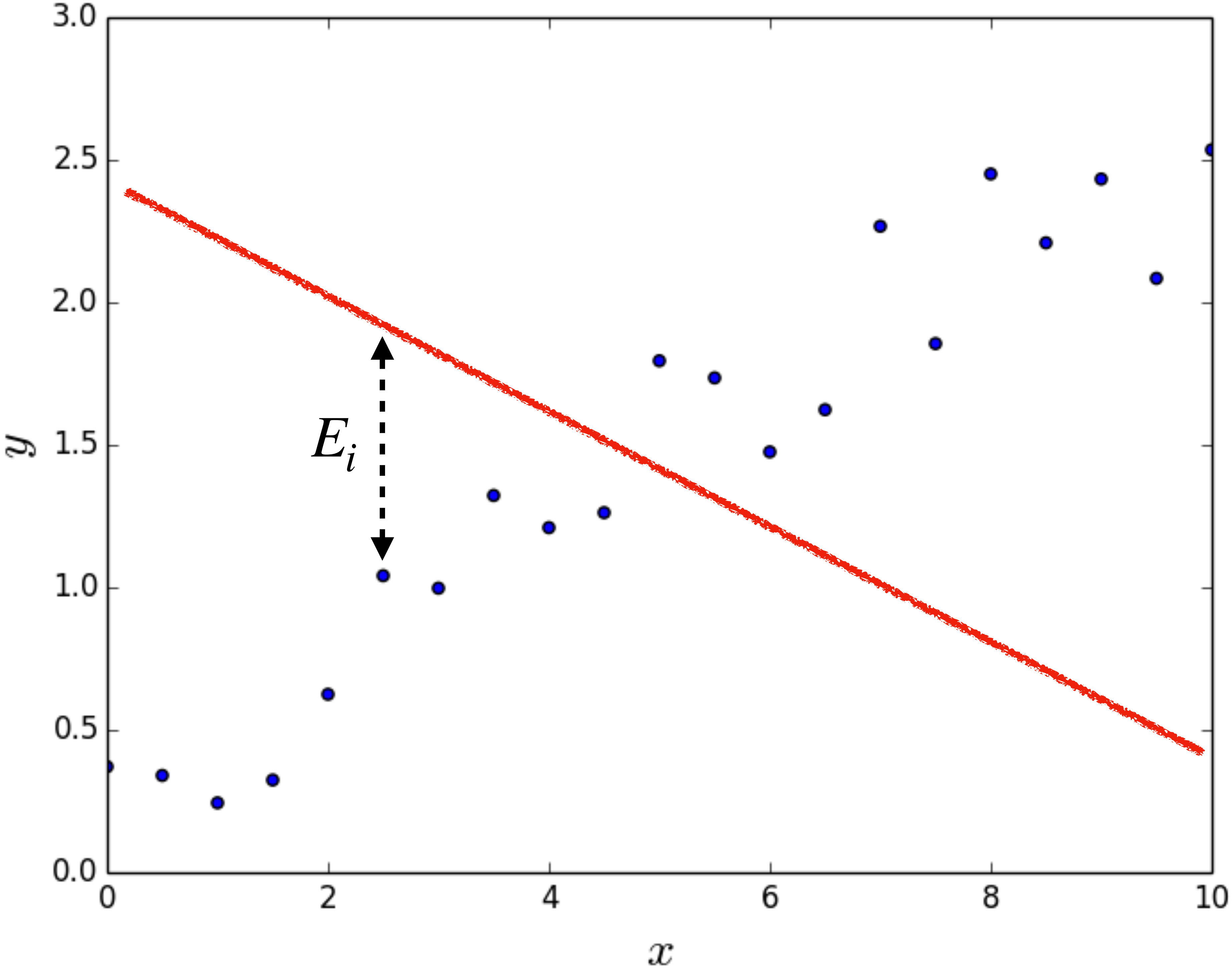
$$y = ax + b$$



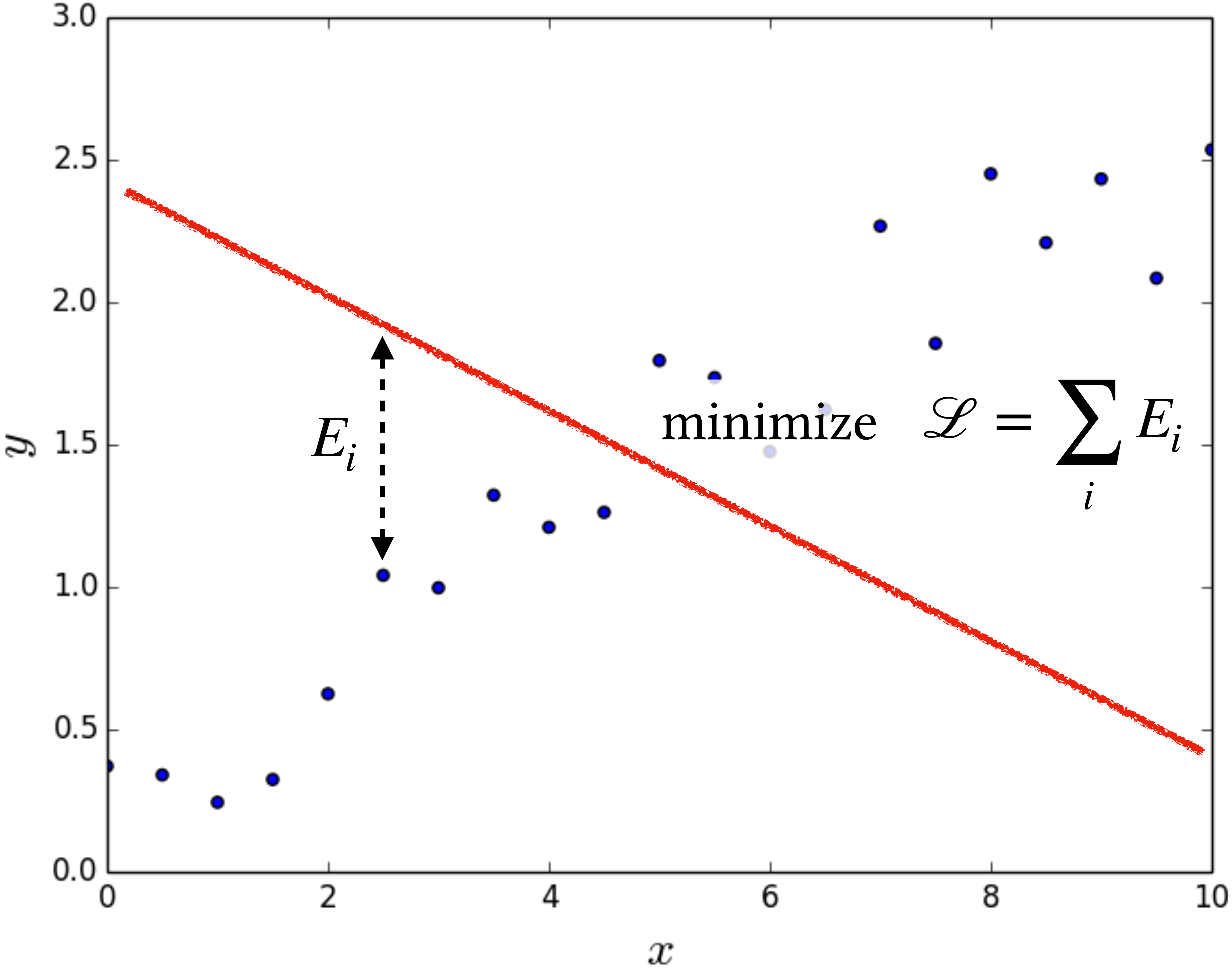
$$y = ax + b$$



$$y = ax + b$$

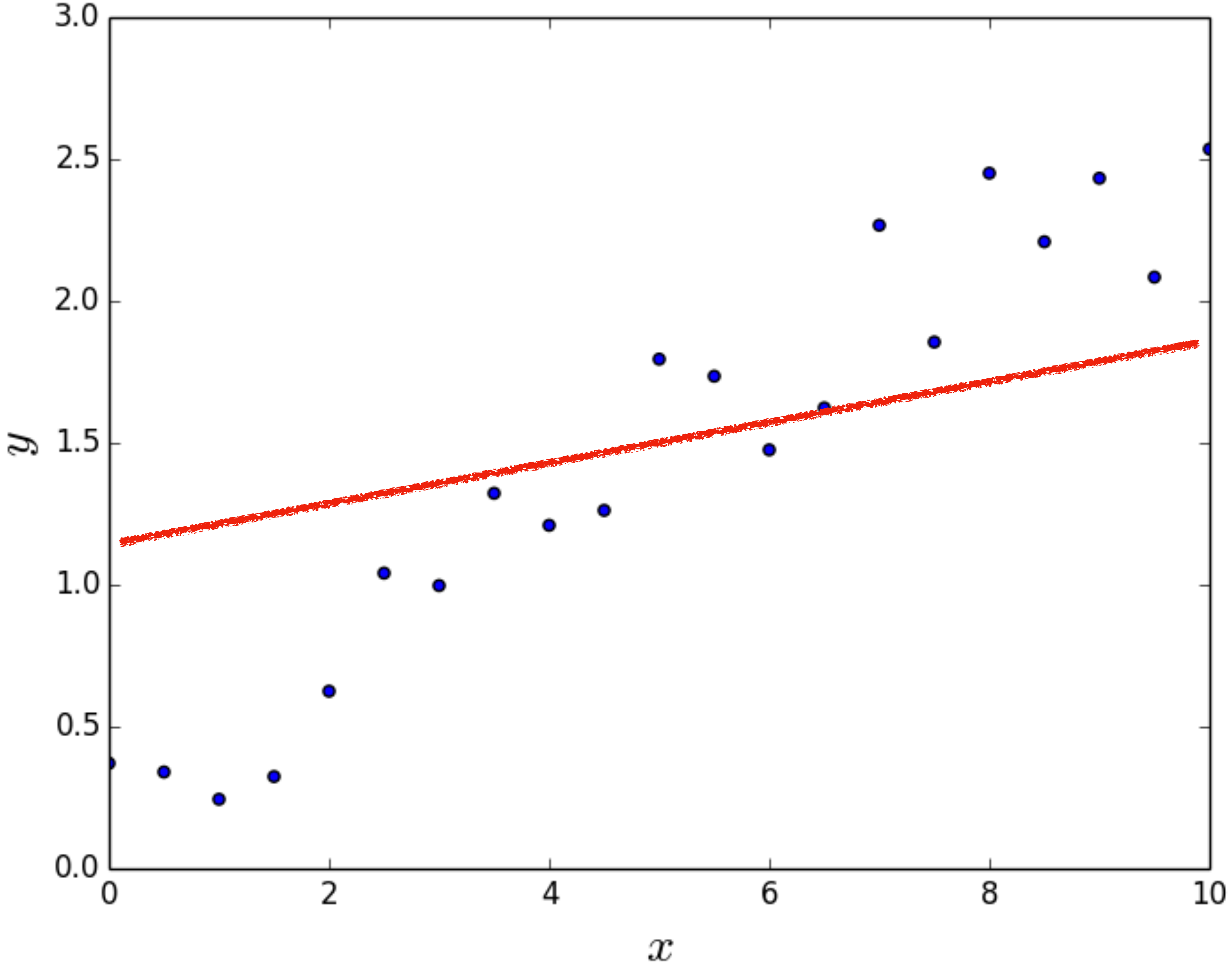


$$y = ax + b$$

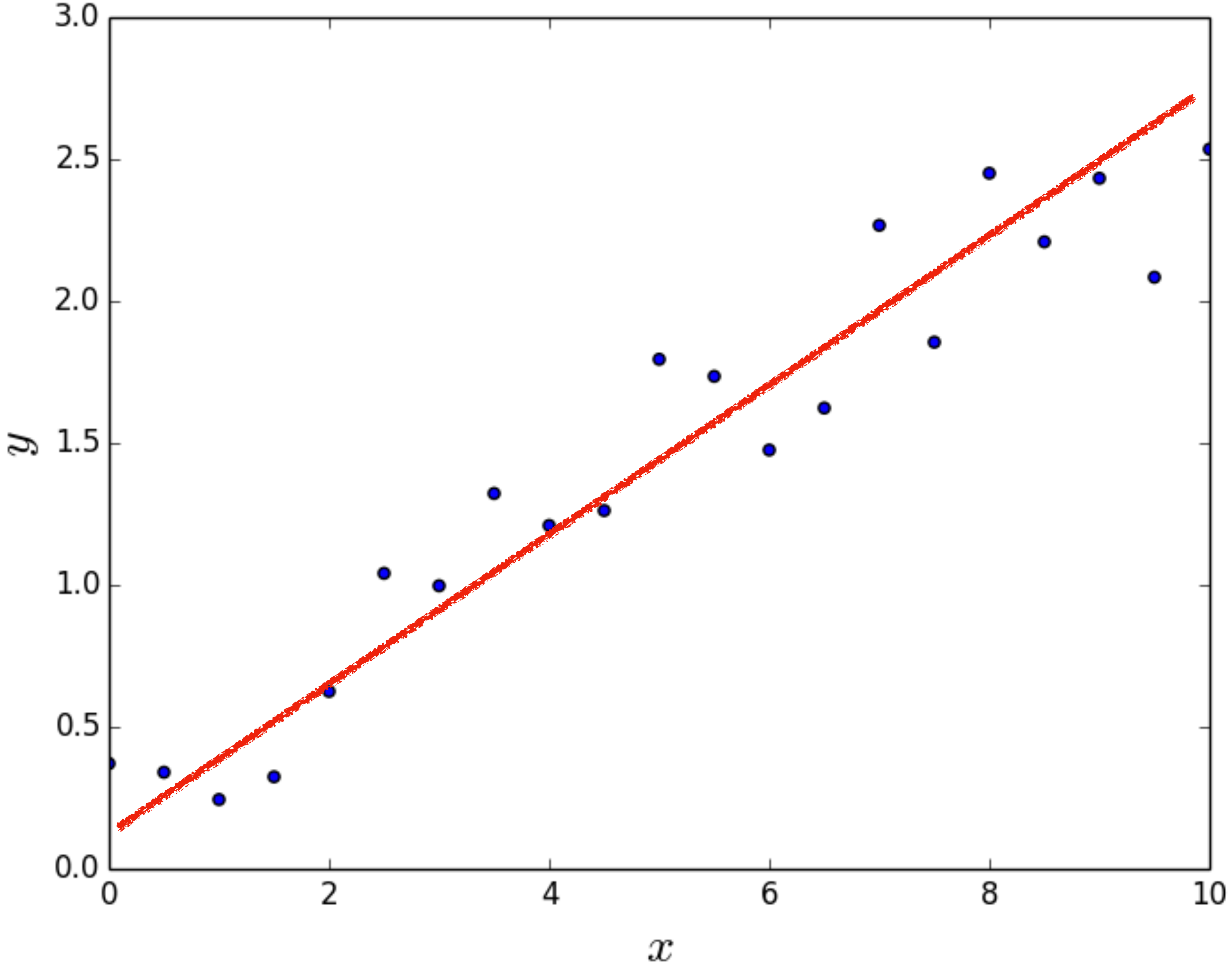


minimize $\mathscr{L} = \sum_i E_i$

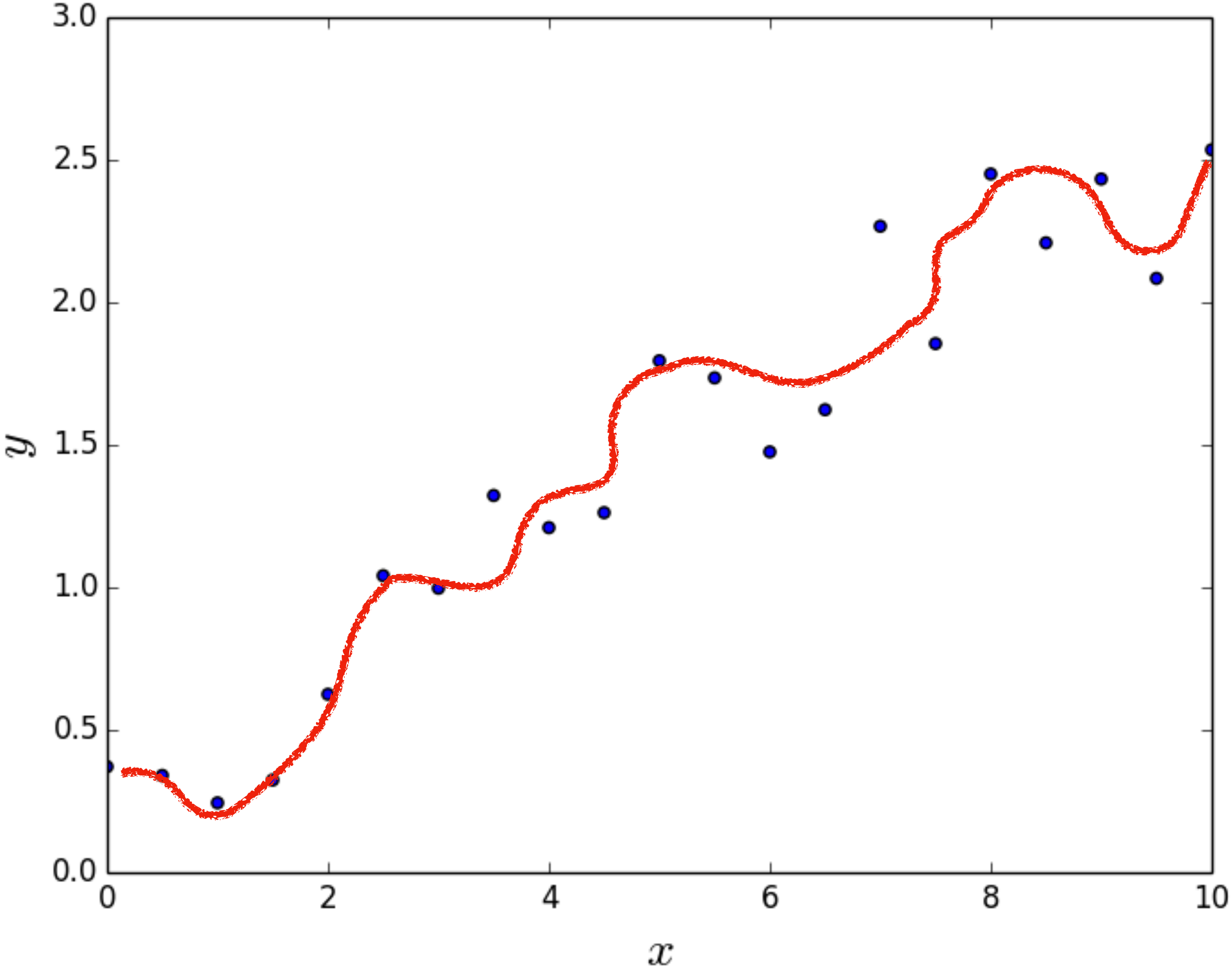
$$y = ax + b$$



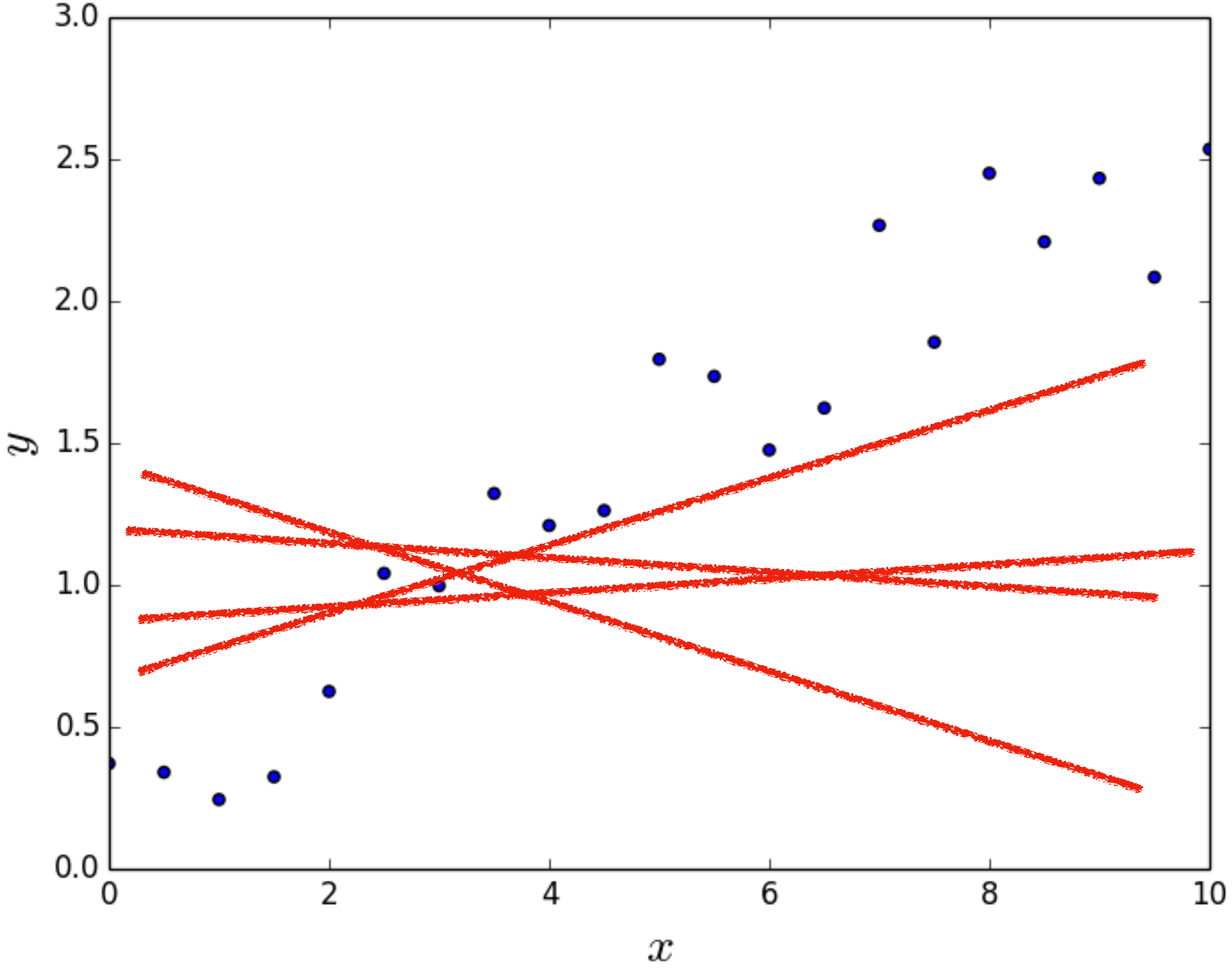
$$y = ax + b$$



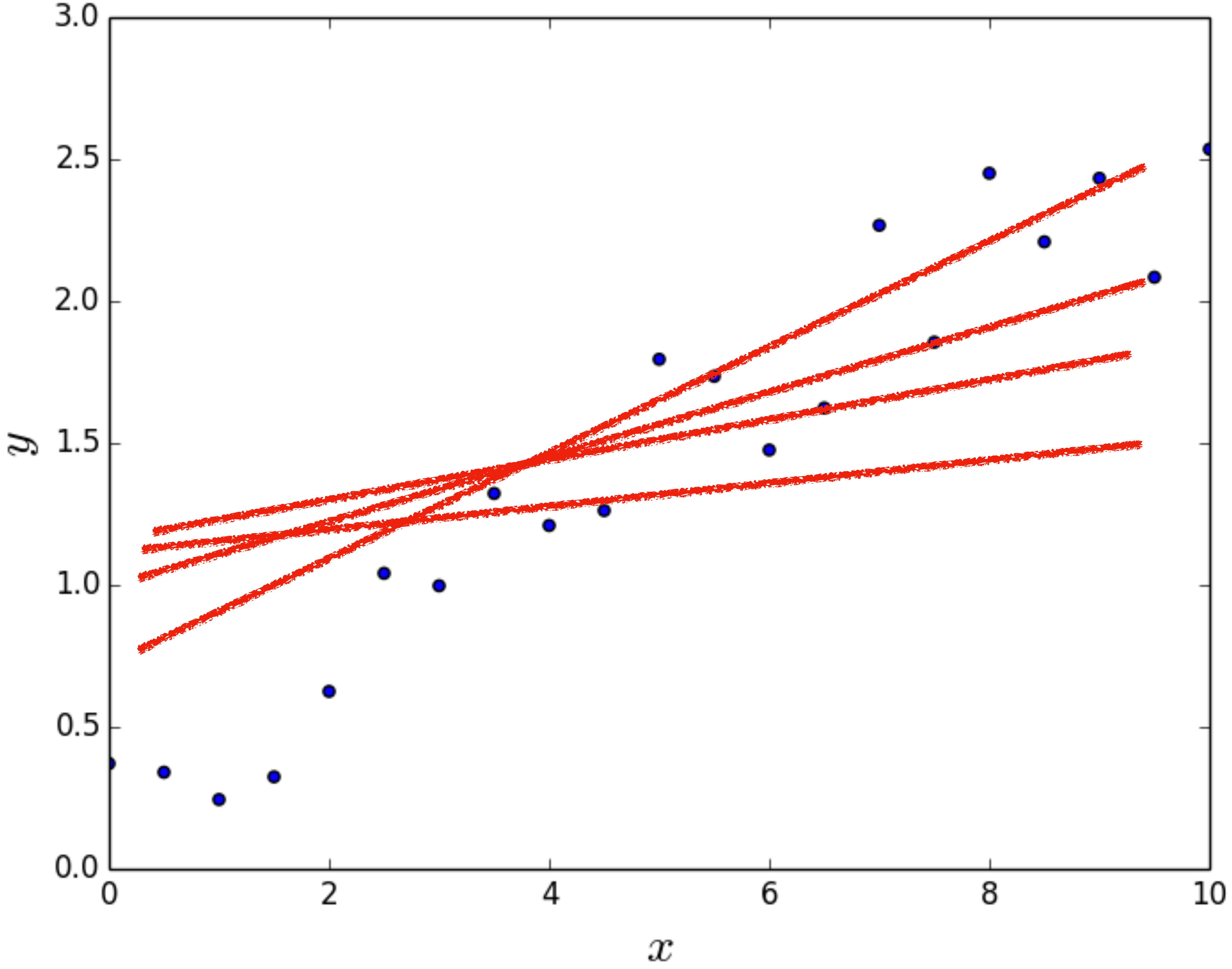
$$y = NN(x)$$



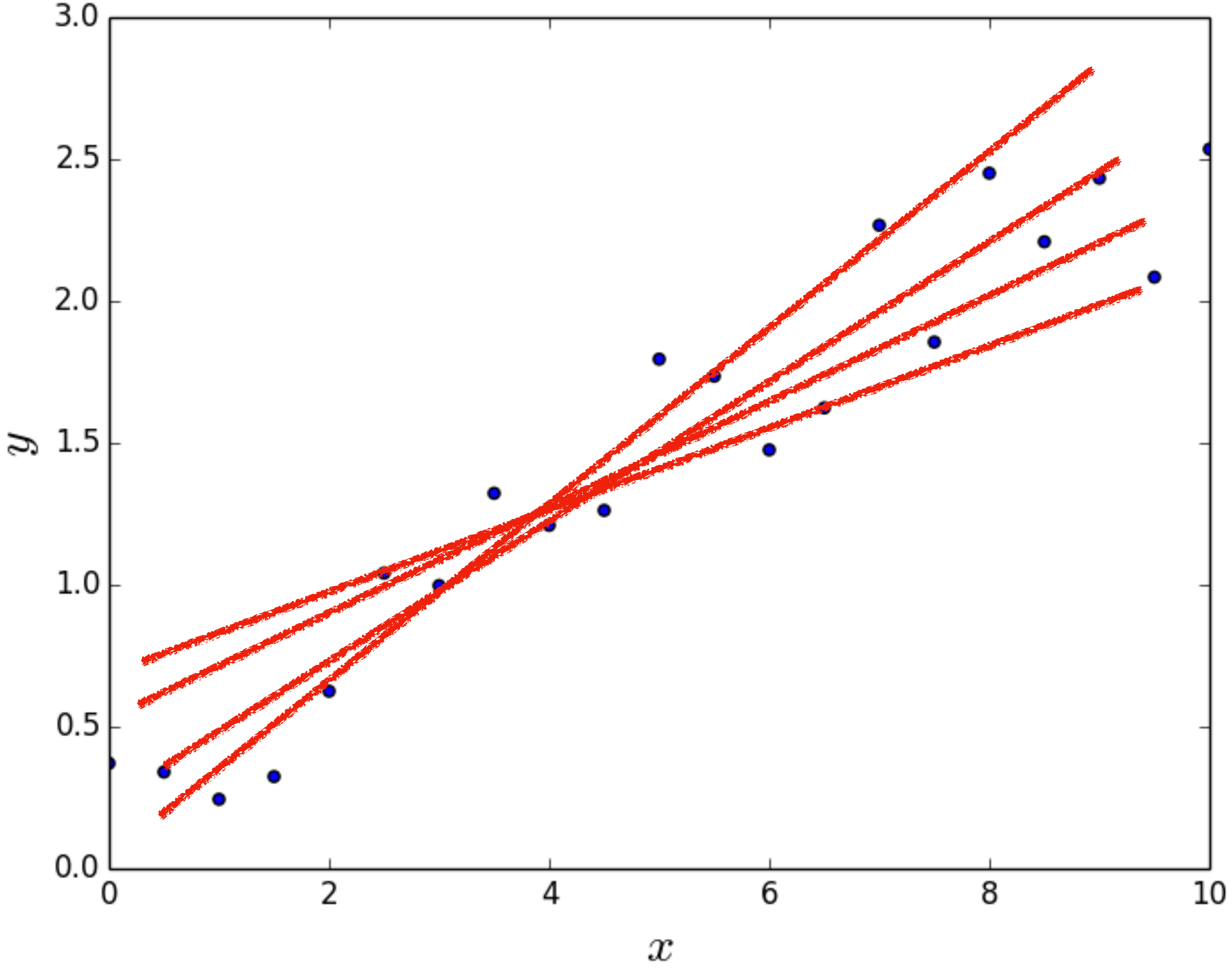
$y = \alpha x + \epsilon$ where $\alpha \sim \mathcal{N}(0,1)$ & $\epsilon \sim \mathcal{N}(1,1)$



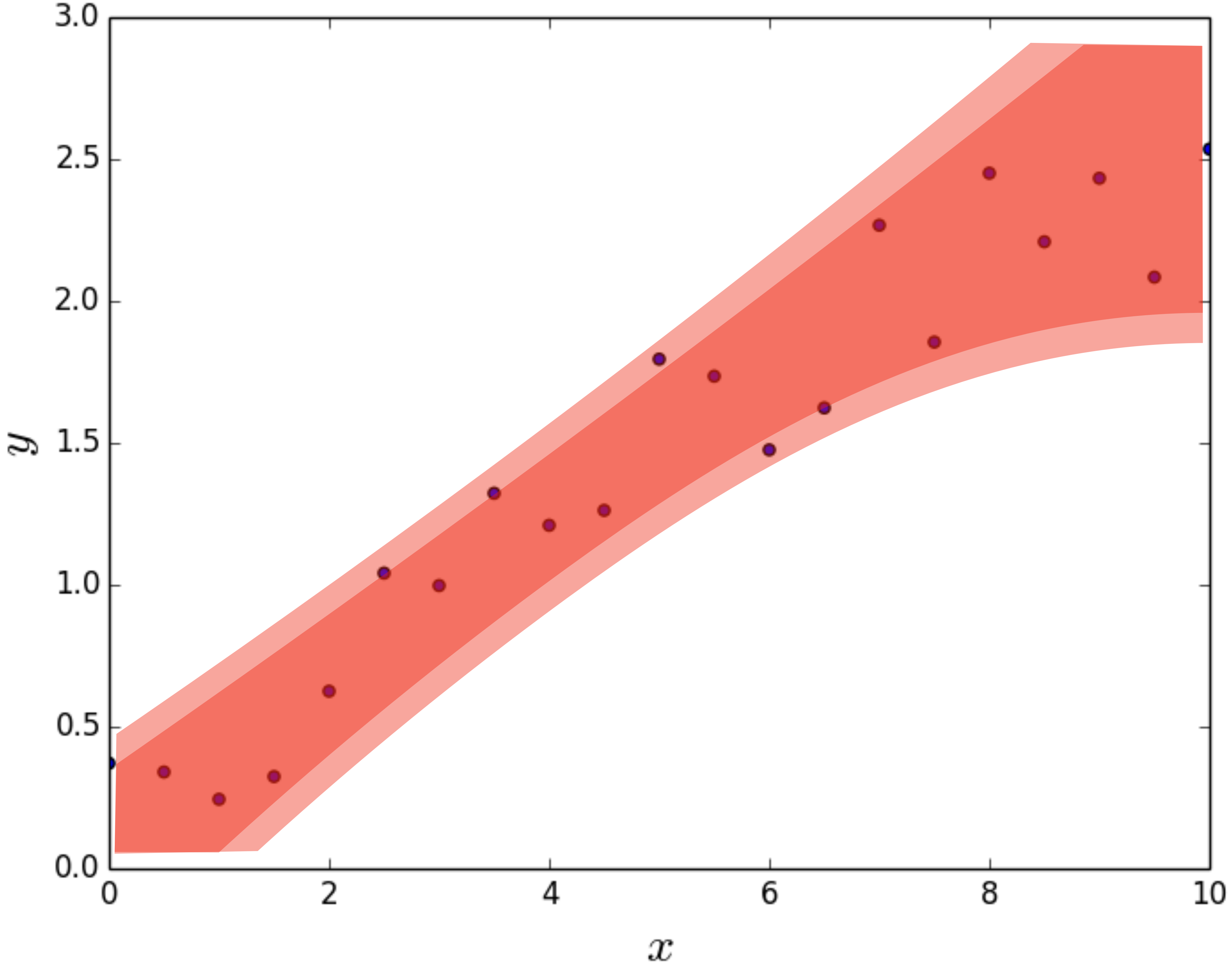
$$y = \alpha x + \epsilon$$

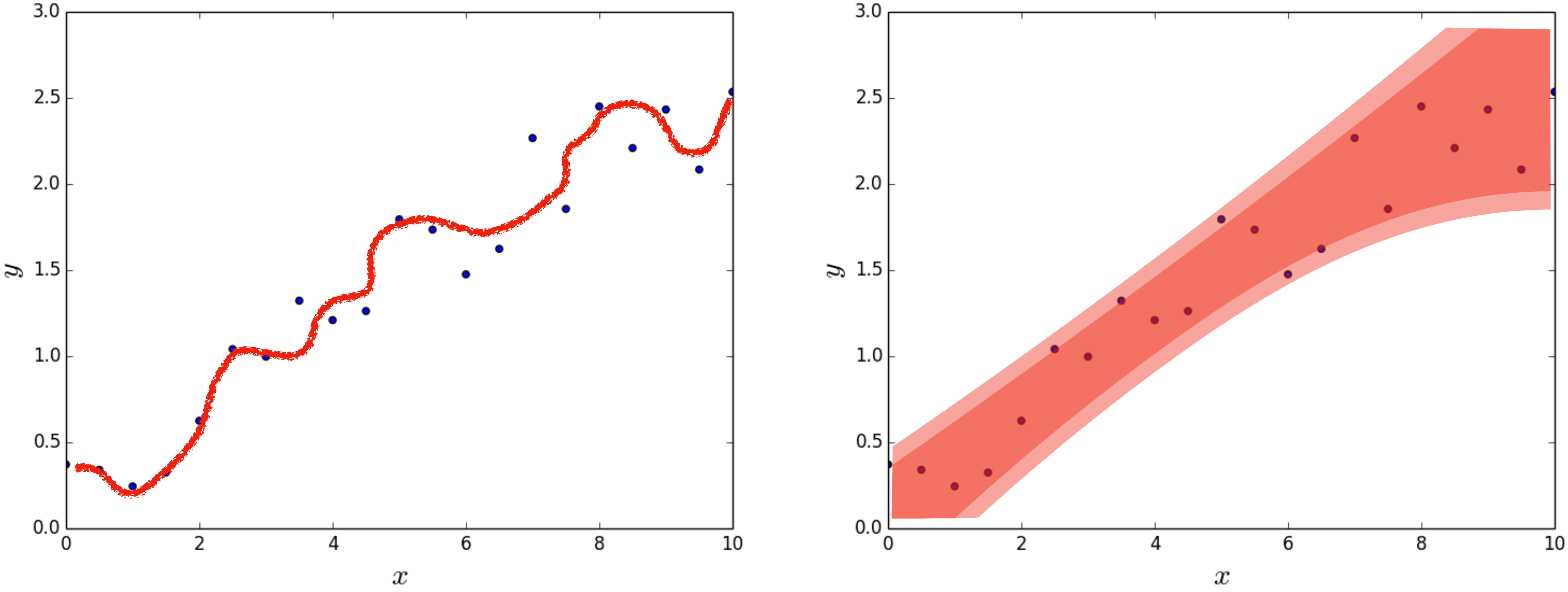


$$y = \alpha x + \epsilon$$



$$y = \alpha x + \epsilon$$




3.0
2.5
2.0
1.5
1.0
0.5
0.0
y
0
2
4
6
8
10
x
3.0
2.5
2.0
1.5
1.0
0.5
0.0
y
0
2
4
6
8
10
x

Input

🤖

Result

Input Weight

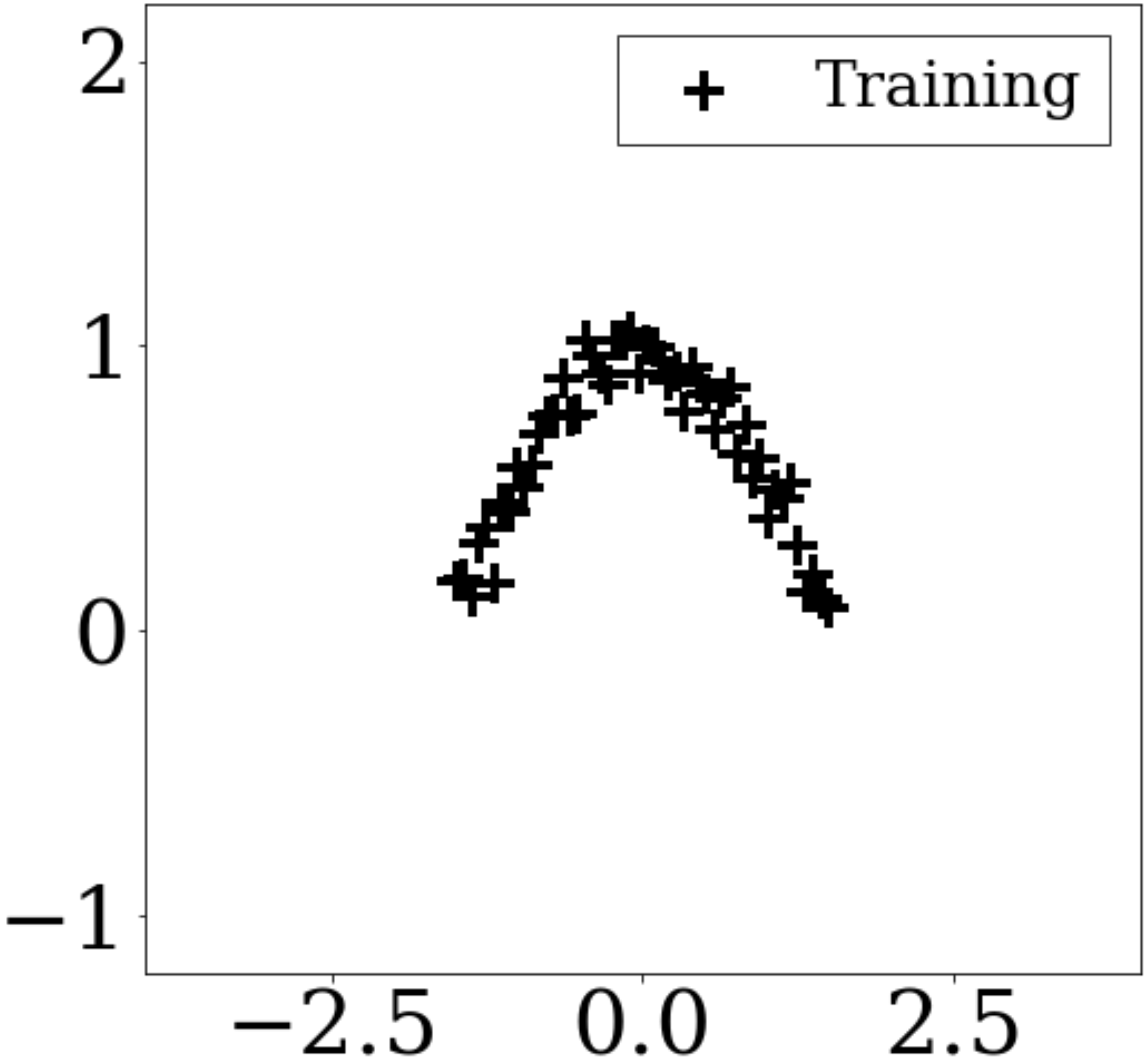
🤖

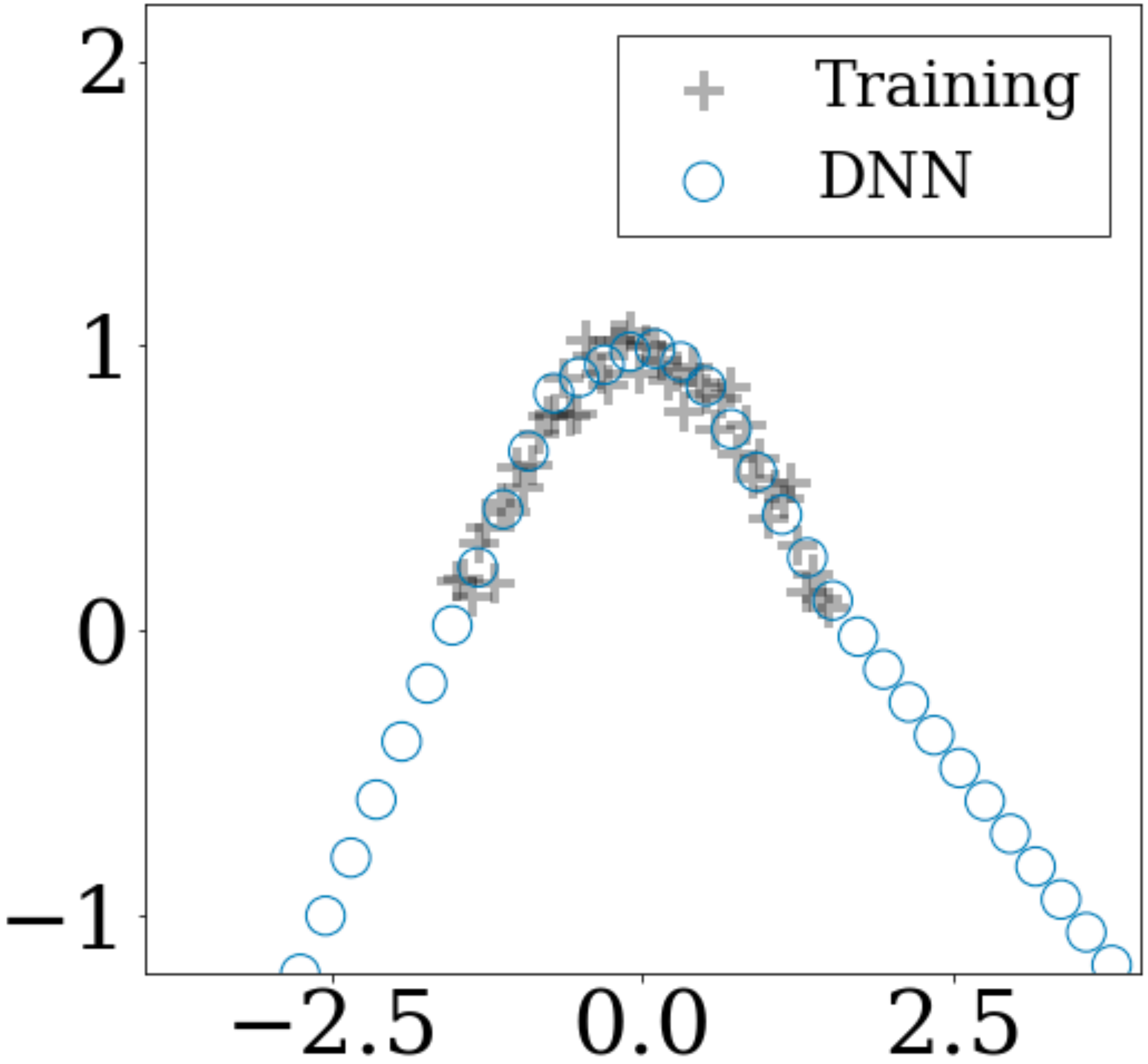
Result

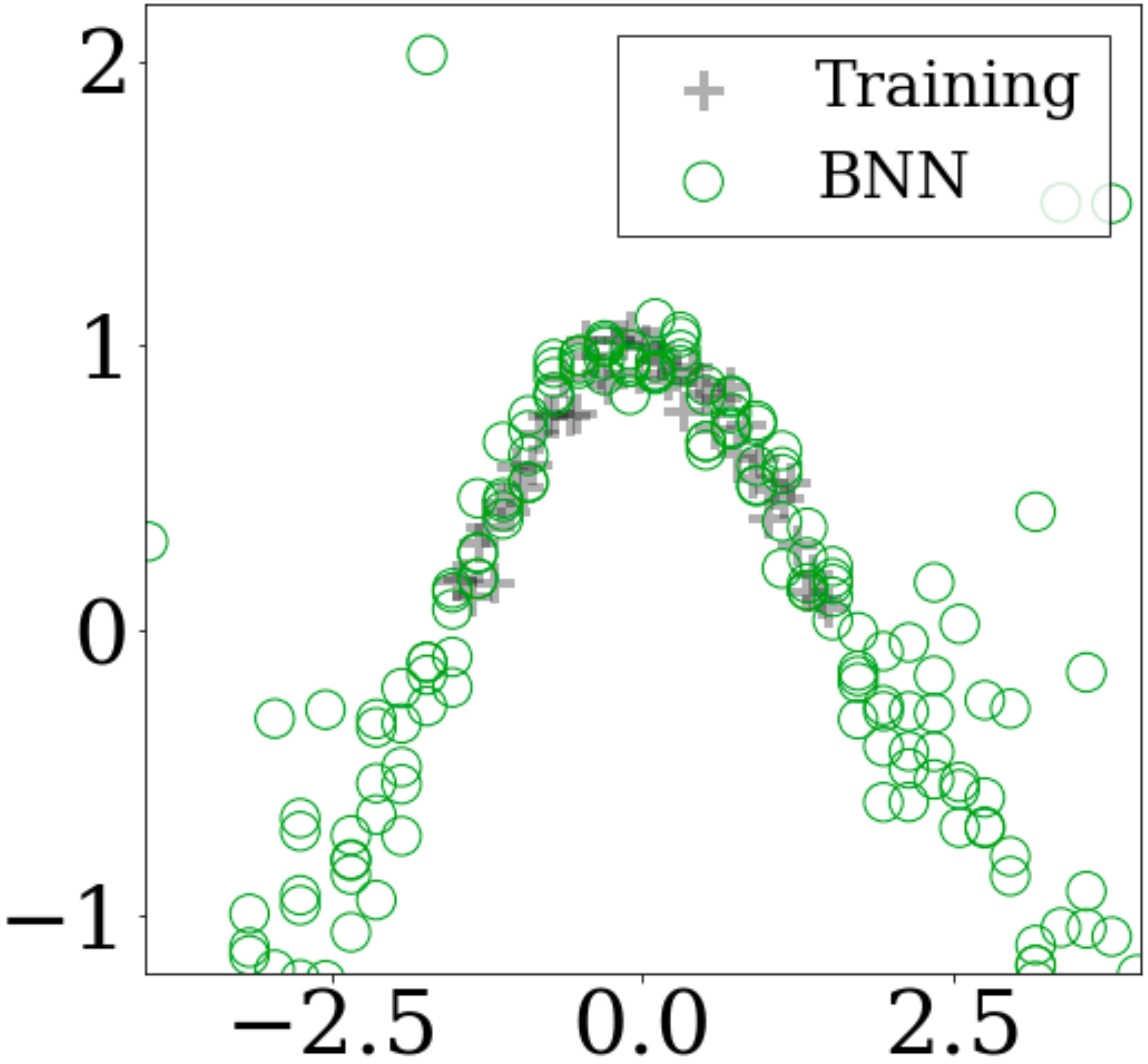
Input Weights

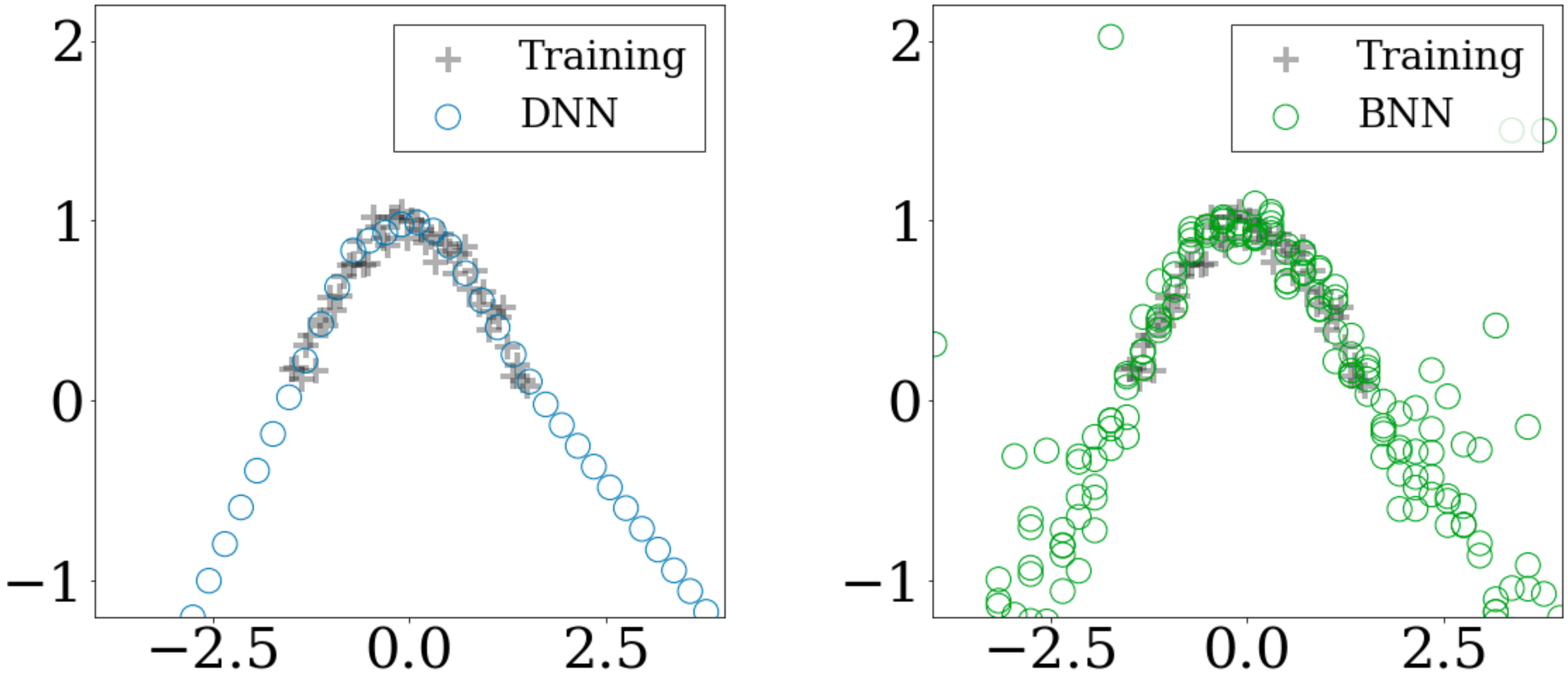
🤖

Results


Training
2
1
0
−1
−2.5
0.0
2.5

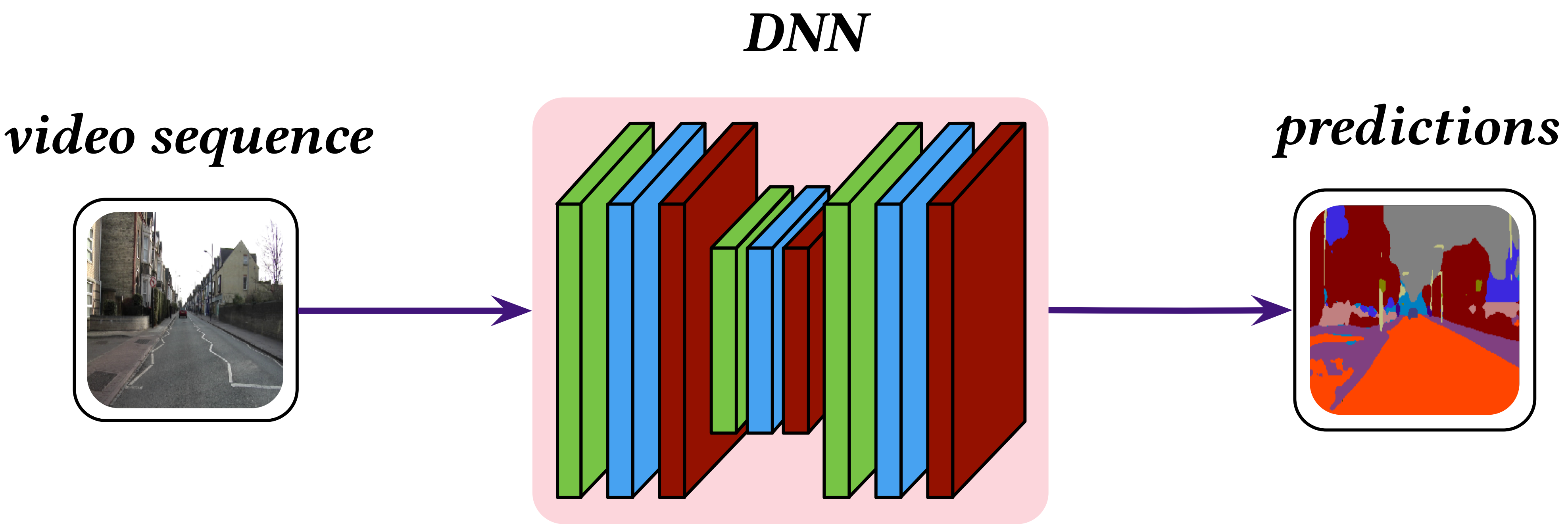

Training
DNN
2
1
0
−1
−2.5
0.0
2.5

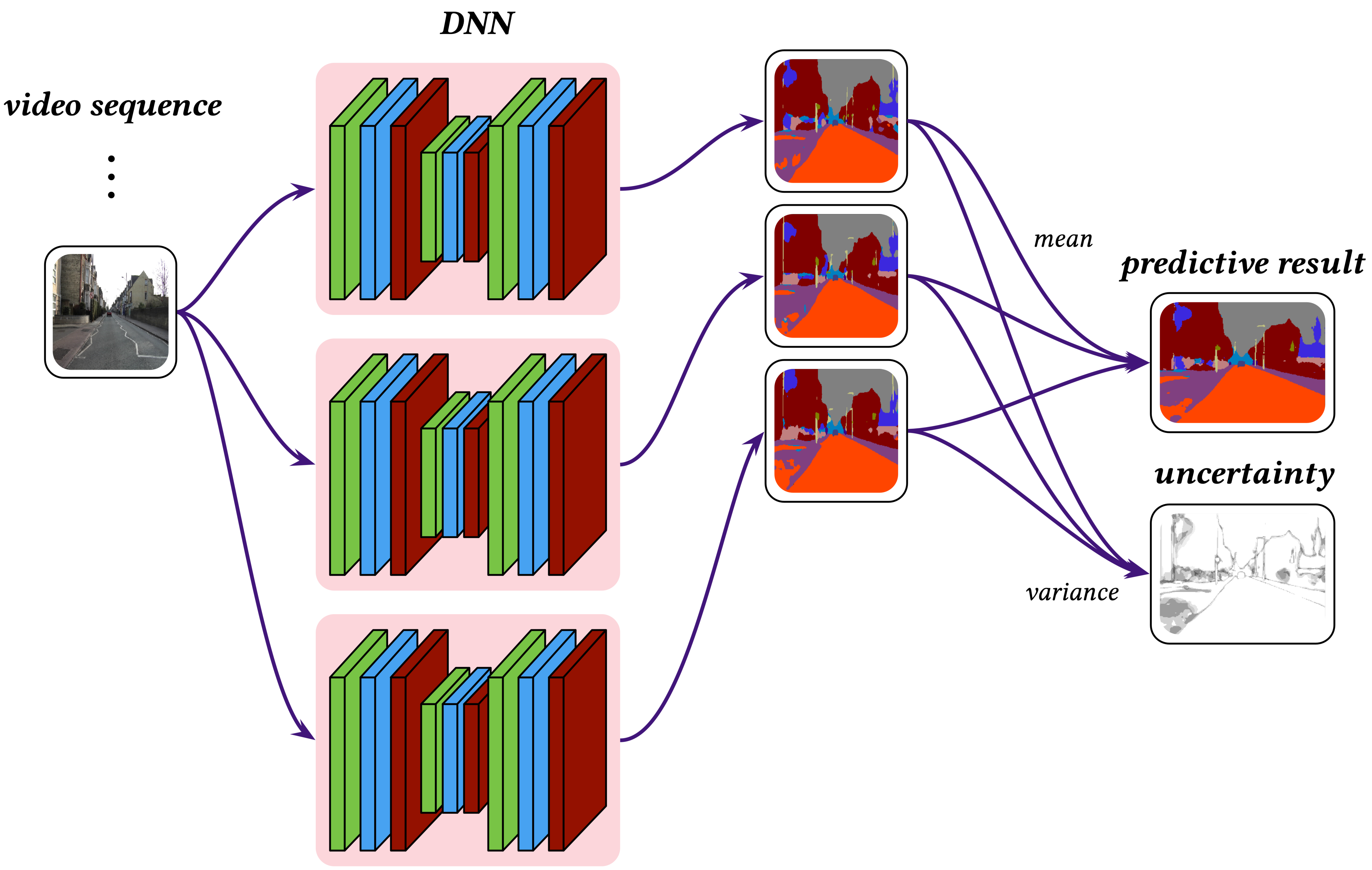

Training
BNN
2
1
0
−1
−2.5
0.0
2.5

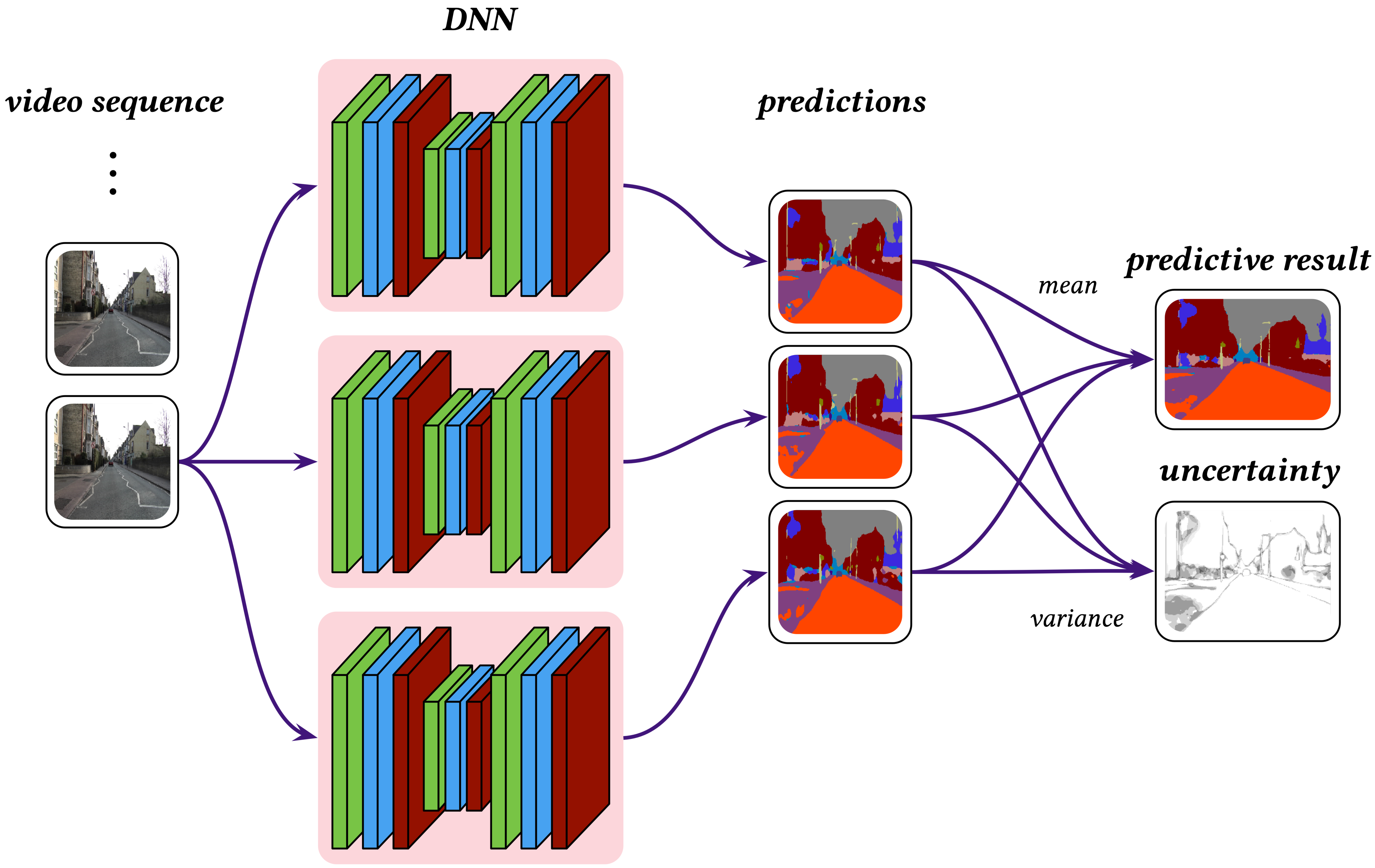

2
1
0
−1
−2.5
0.0
2.5
Training
DNN
2
1
0
−1
−2.5
0.0
2.5
Training
BNN

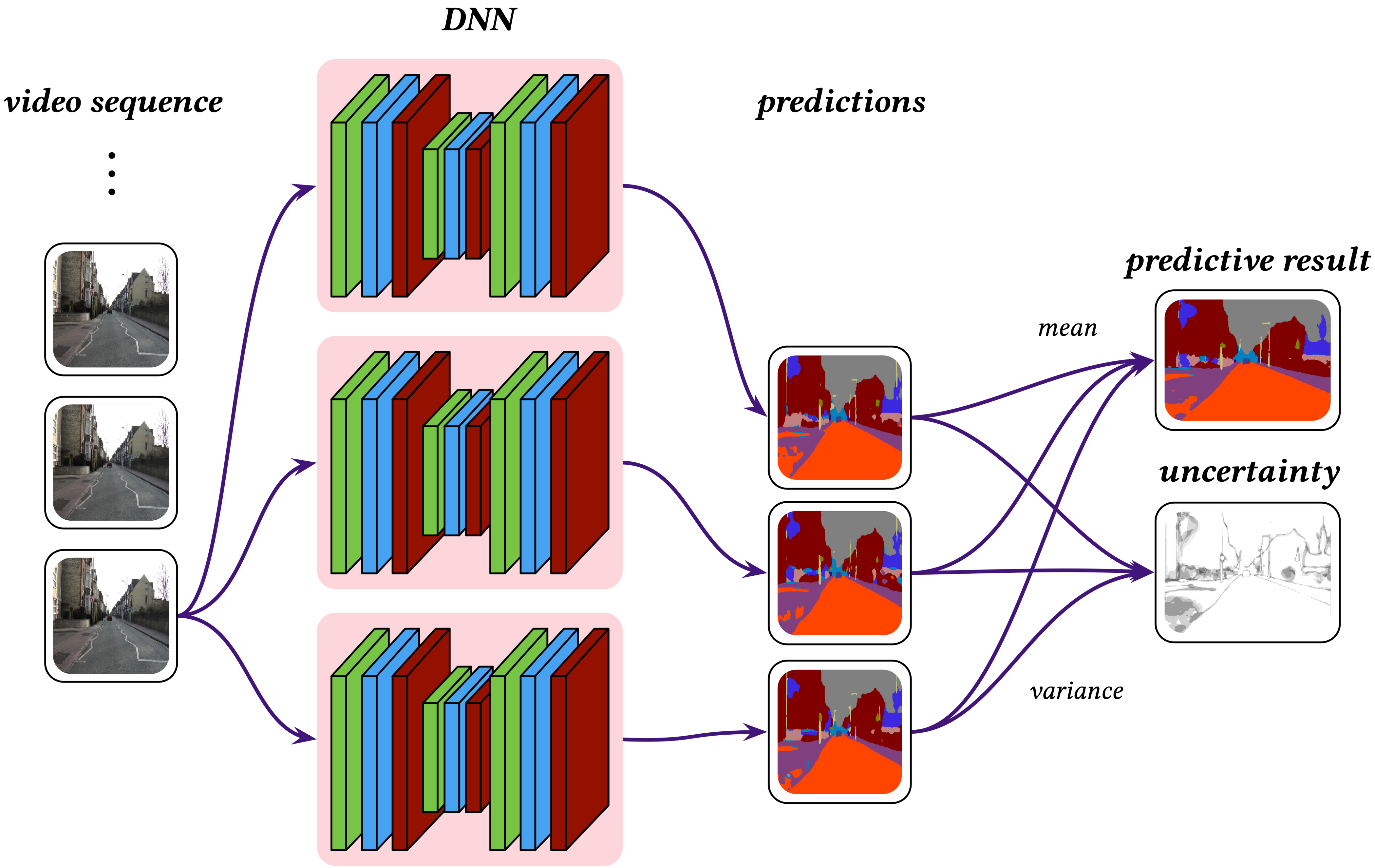
Domino's

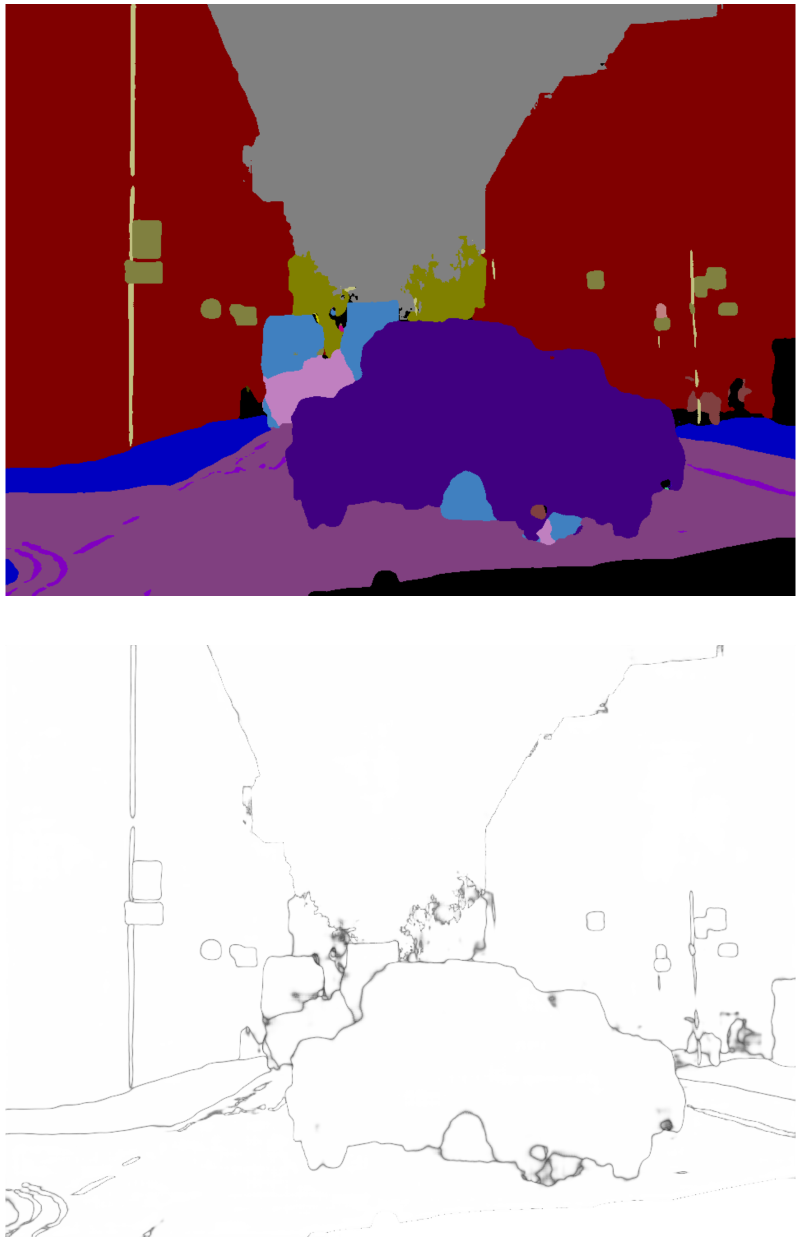
Domino's


DNN
video sequence
predictions

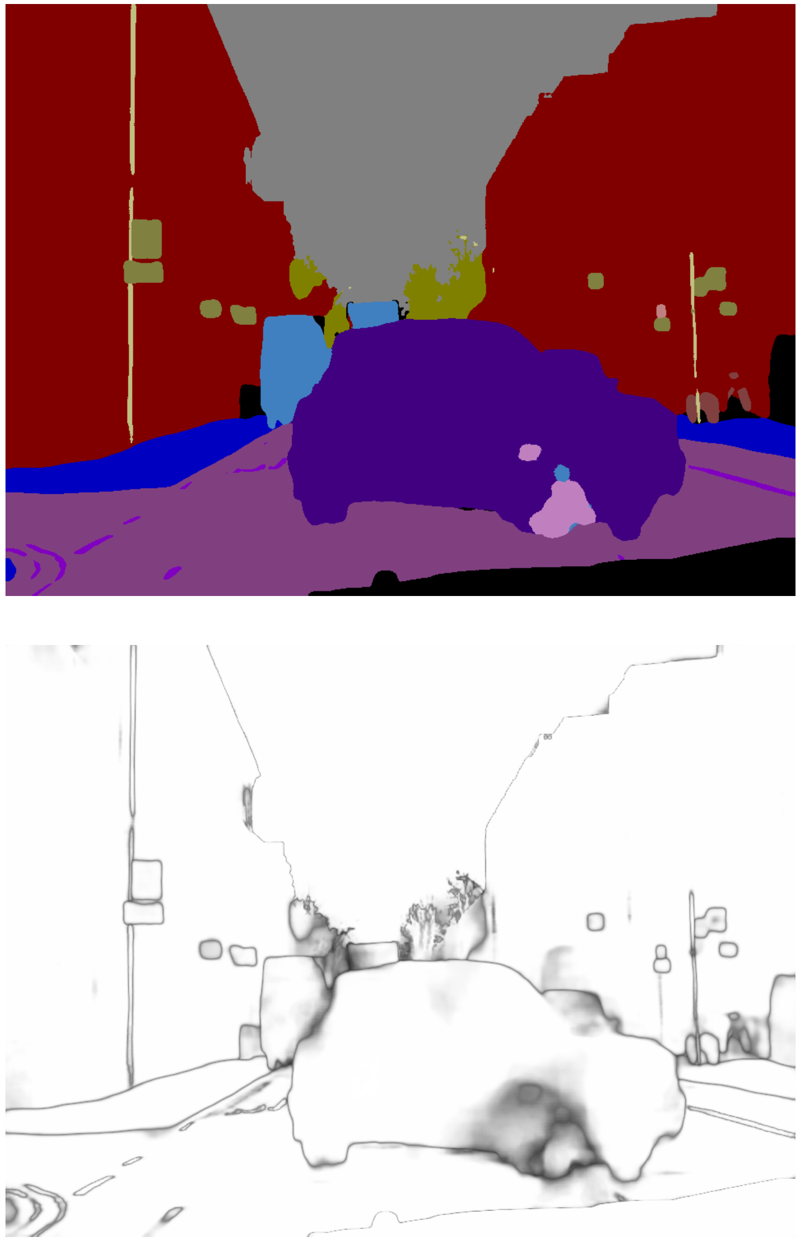

DNN
video sequence
mean
predictive result
uncertainty
variance


DNN
video sequence
predictions
predictive result
mean
uncertainty
variance

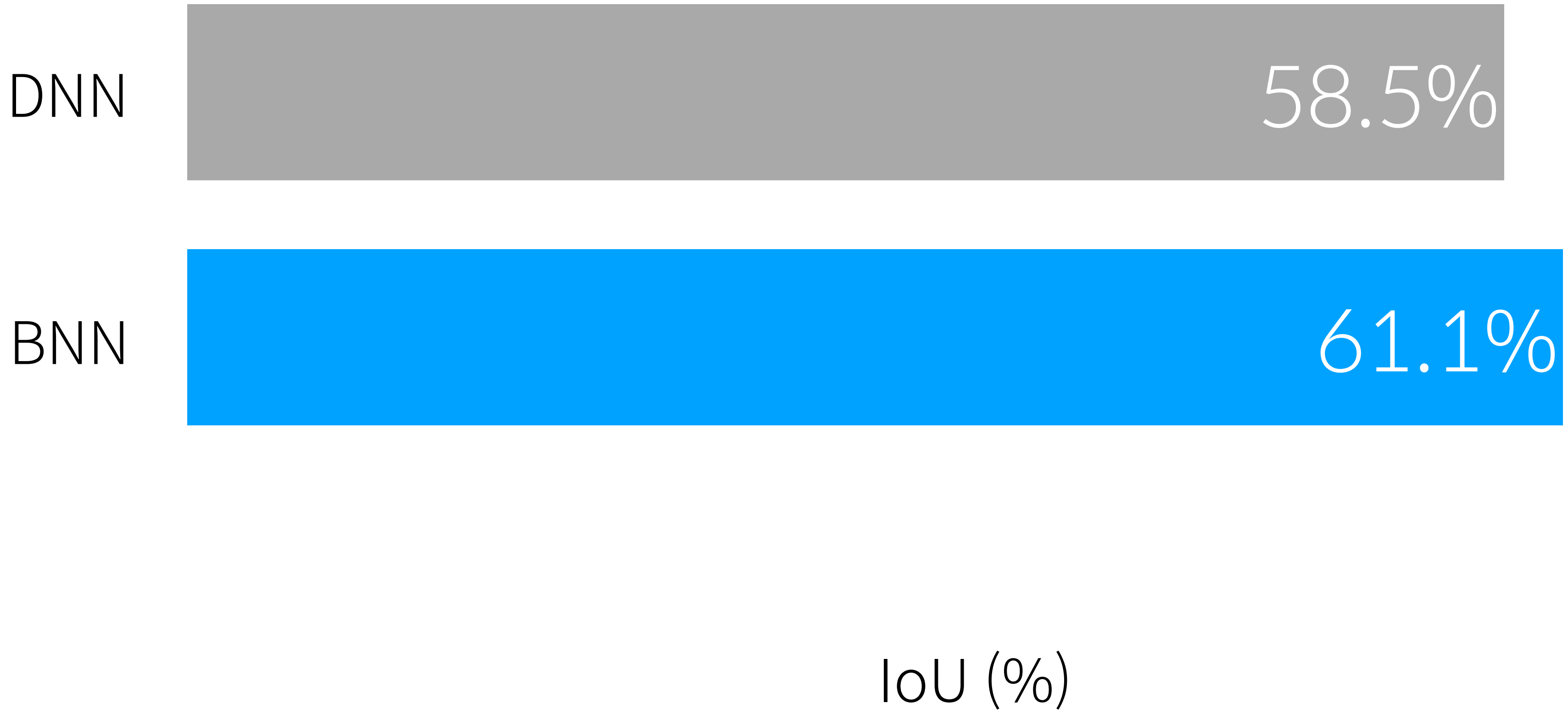

DNN
video sequence
predictions
predictive result
mean
uncertainty
variance

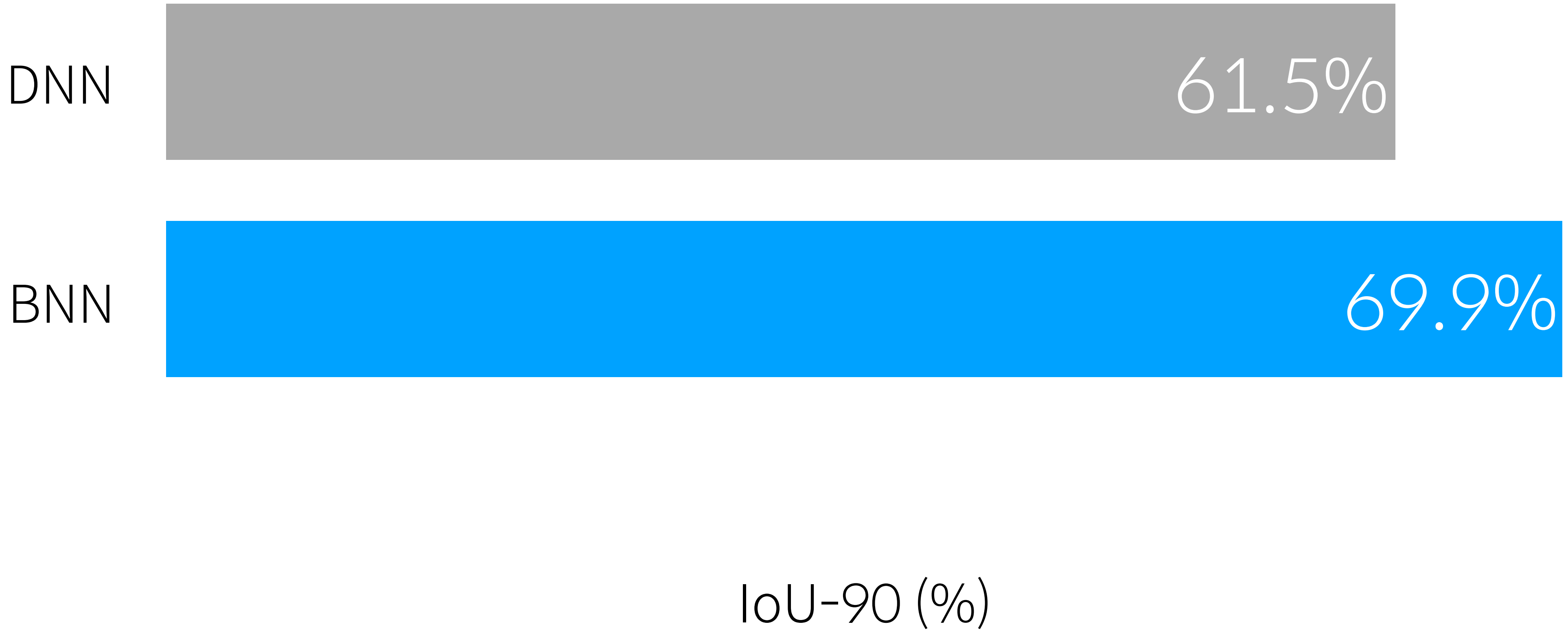


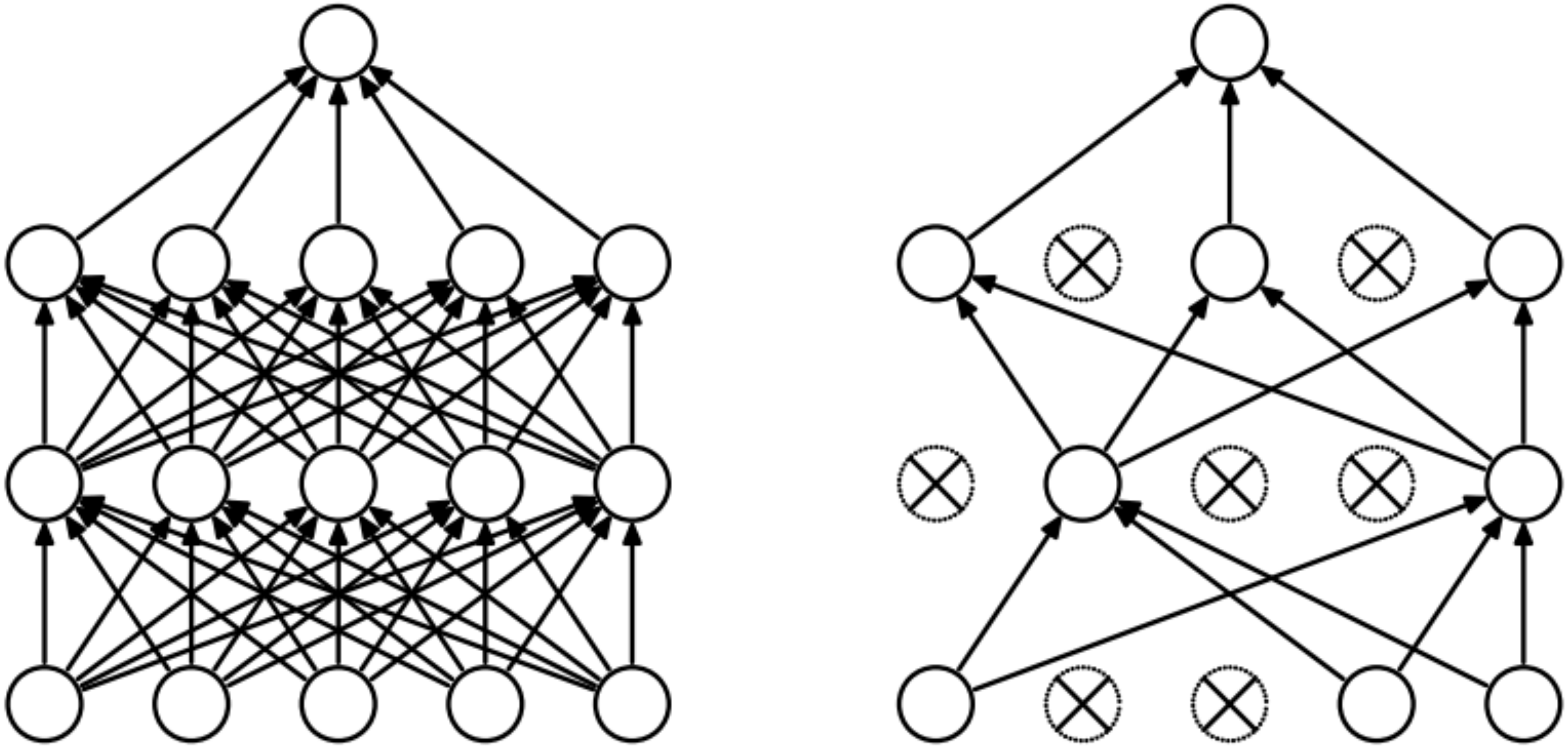
DNN



BNN


DNN
58.5%
BNN
61.1%
IoU (%)


DNN
61.5%
BNN
69.9%
IoU-90 (%)



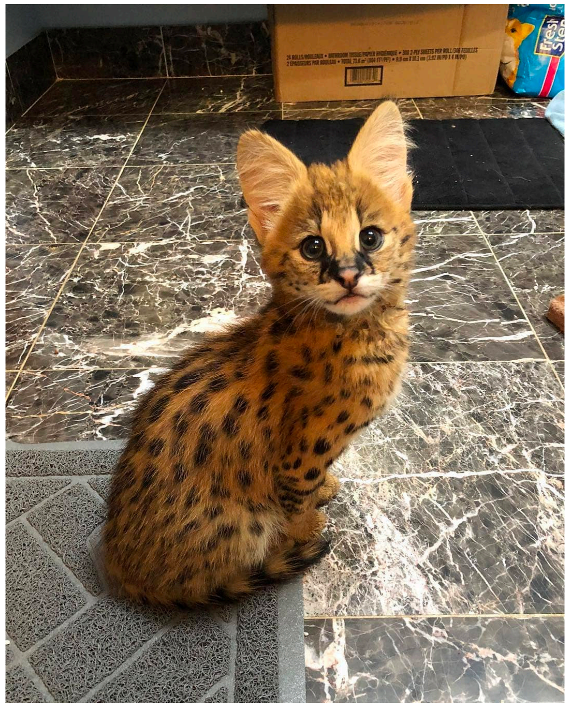
(a) Standard Neural Net

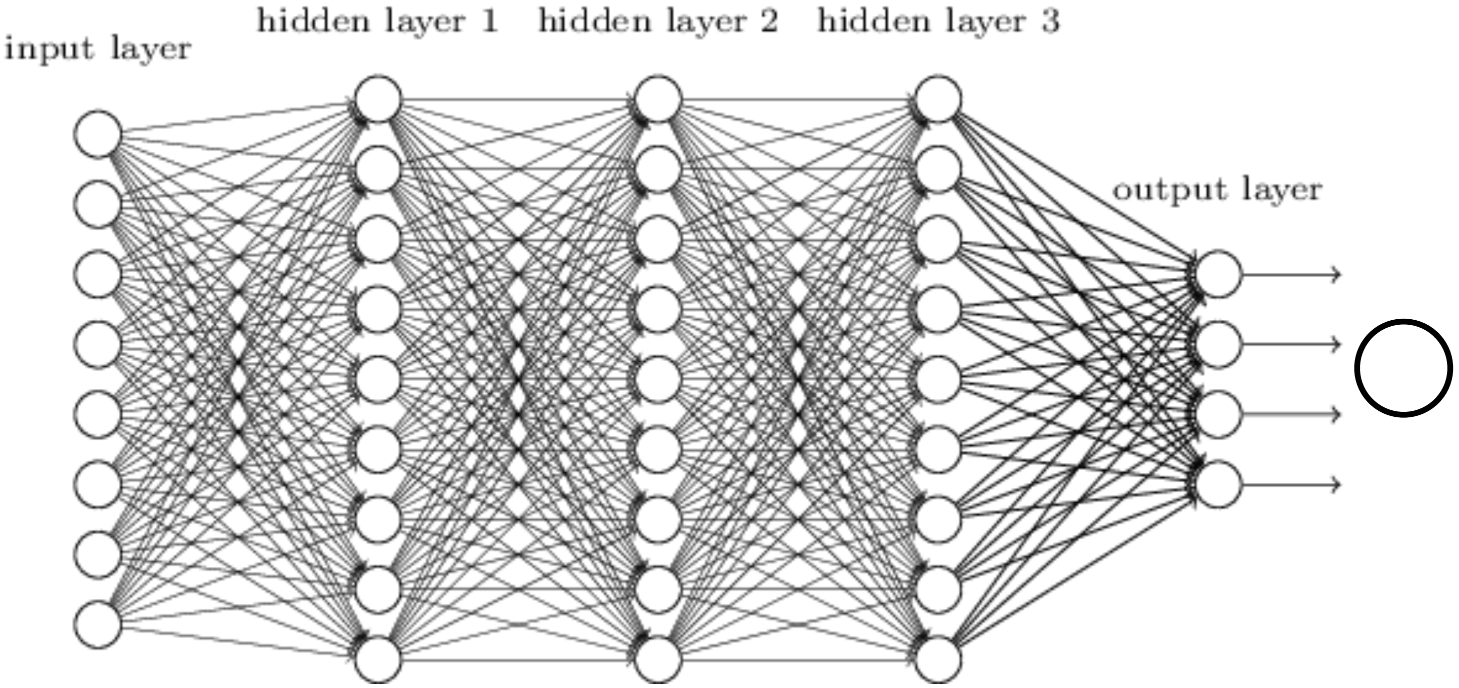
(b) After applying dropout.

input image

deep neural network

output result



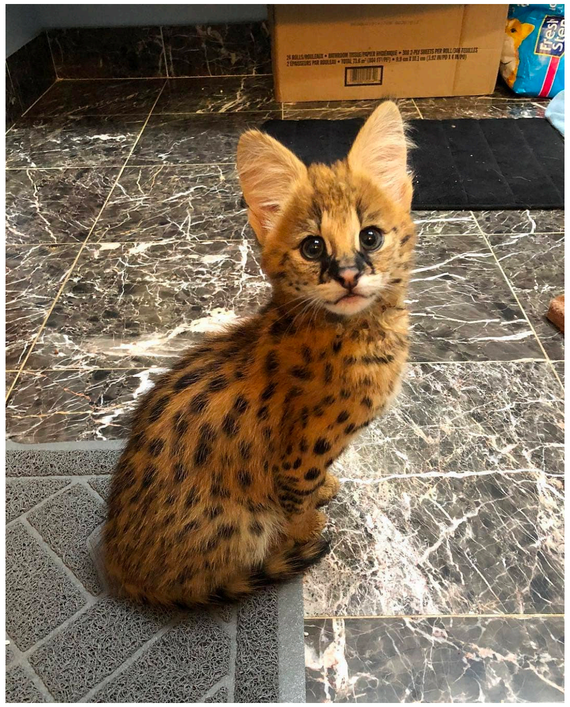


Dog?

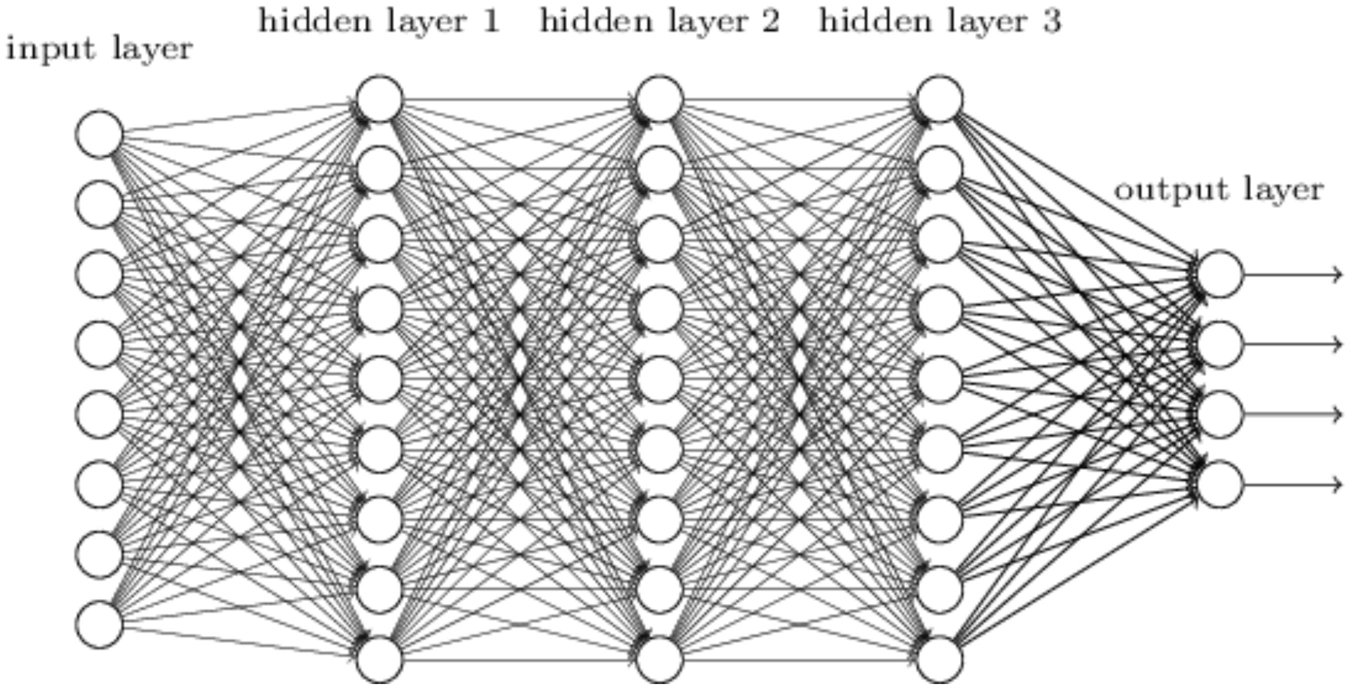
Cat?

http://neuralnetworksanddeeplearning.com/chap6.html

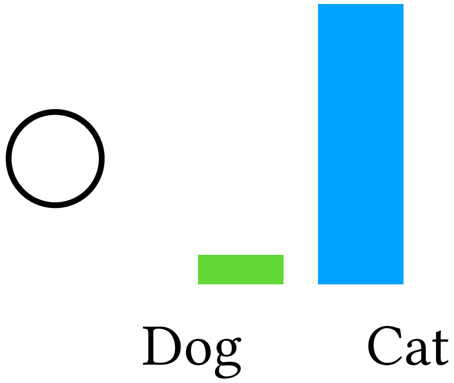
input image



deep neural network



output result

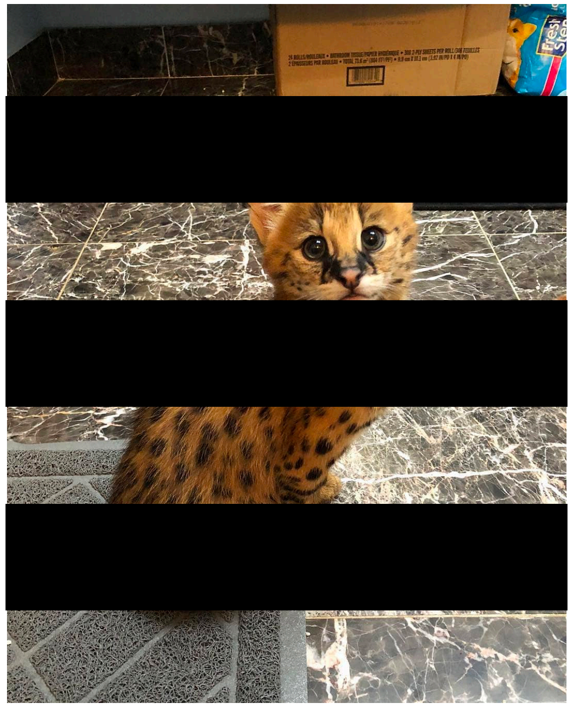


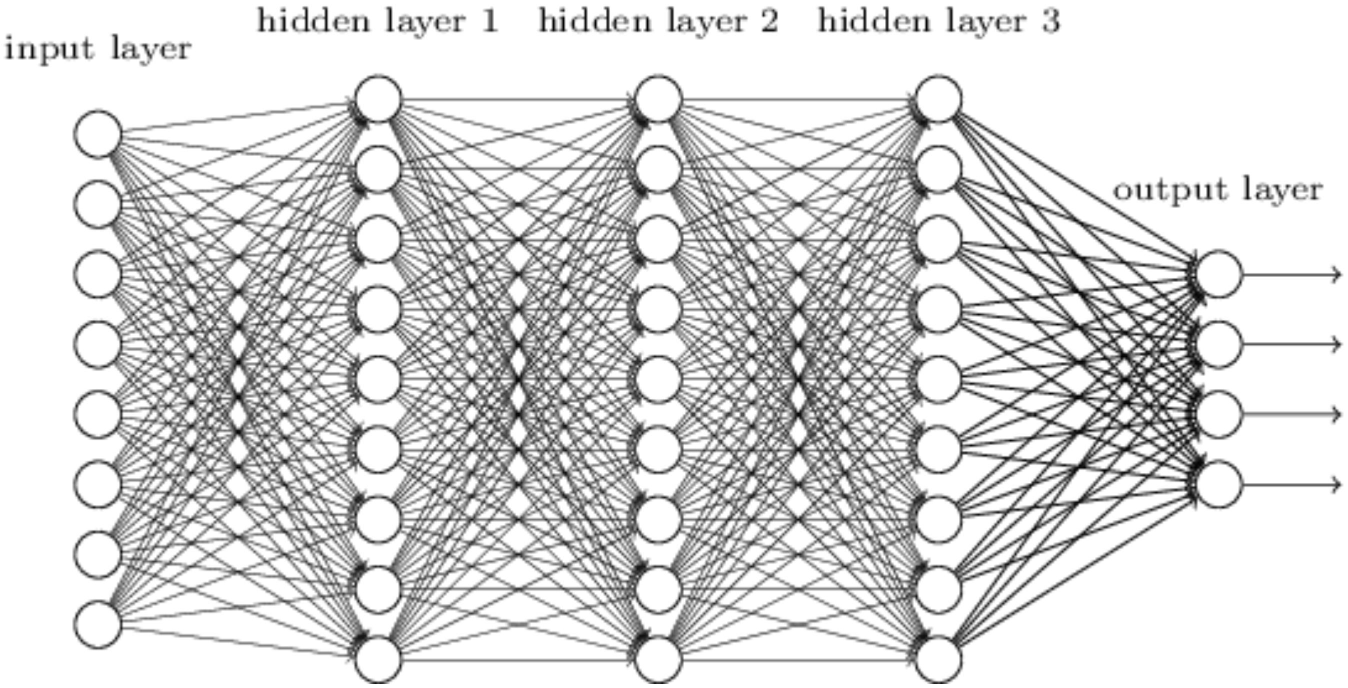
http://neuralnetworksanddeeplearning.com/chap6.html

input image

deep neural network

output result





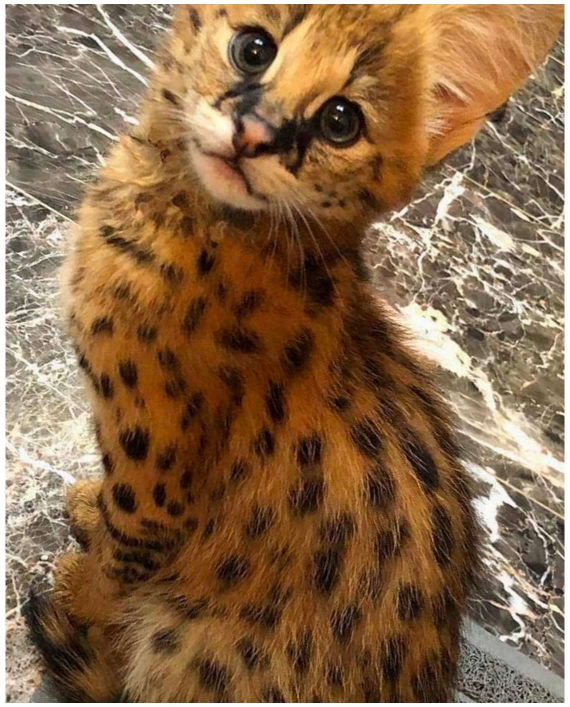
○

Cat?

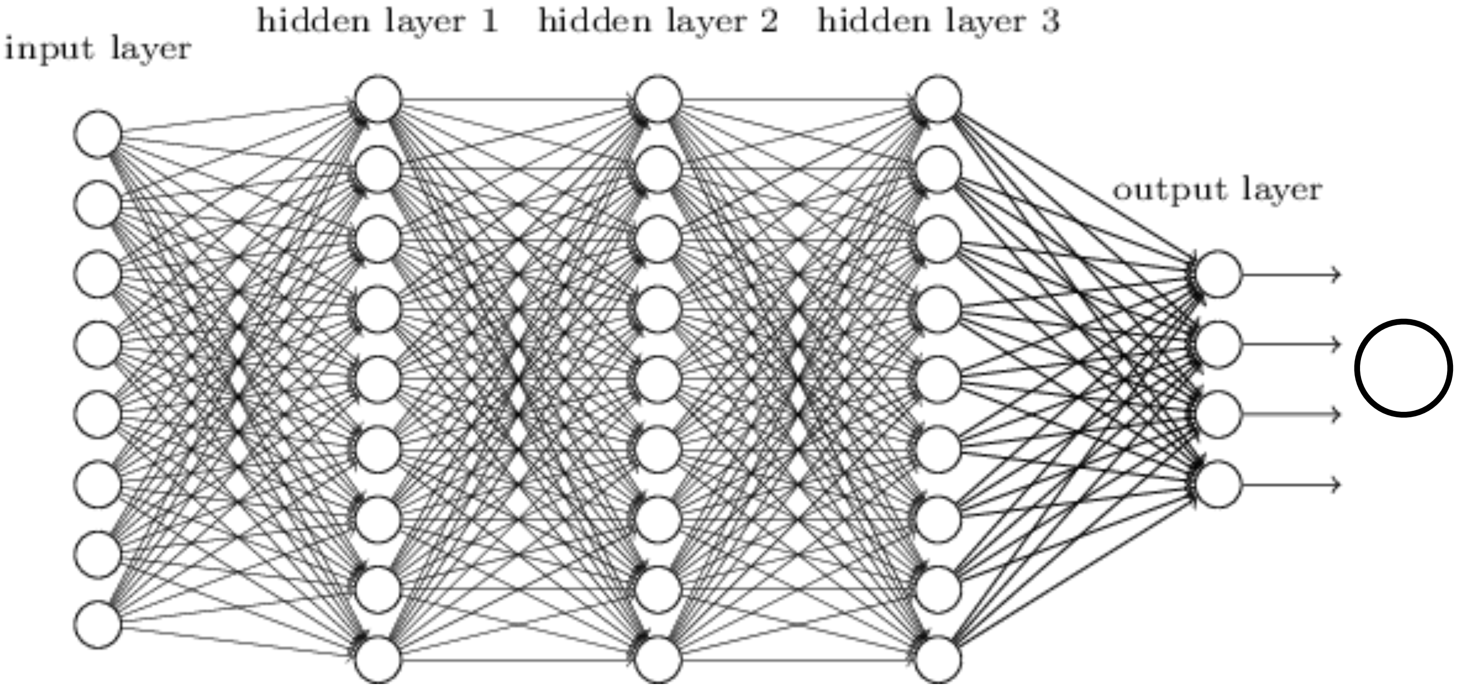
Dog?

http://neuralnetworksanddeeplearning.com/chap6.html

input image



deep neural network



output result

Cat?

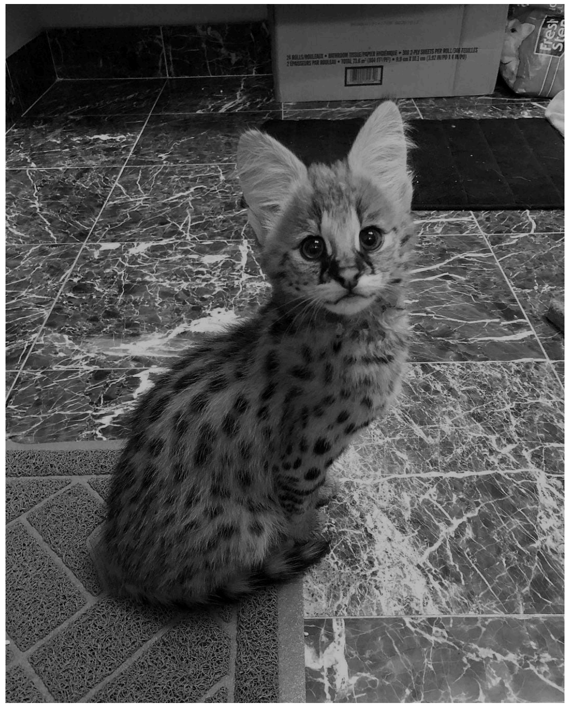
Dog?

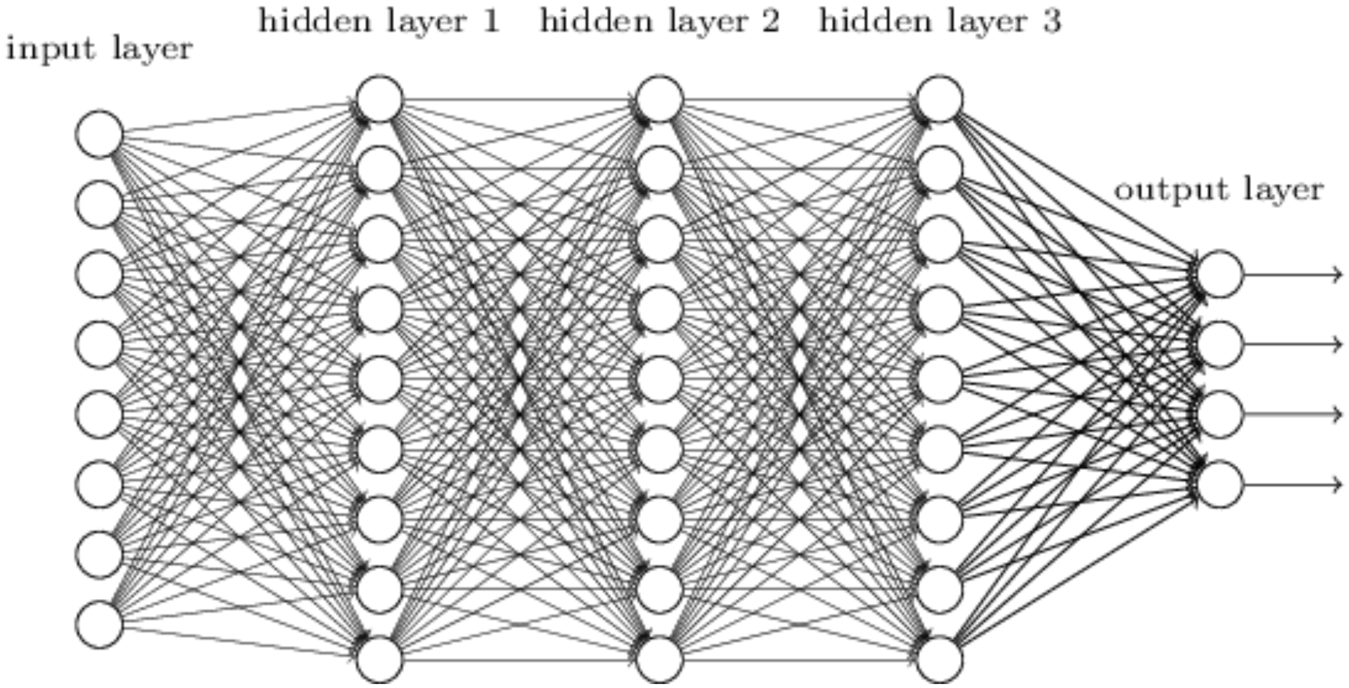
http://neuralnetworksanddeeplearning.com/chap6.html

input image

deep neural network

output result

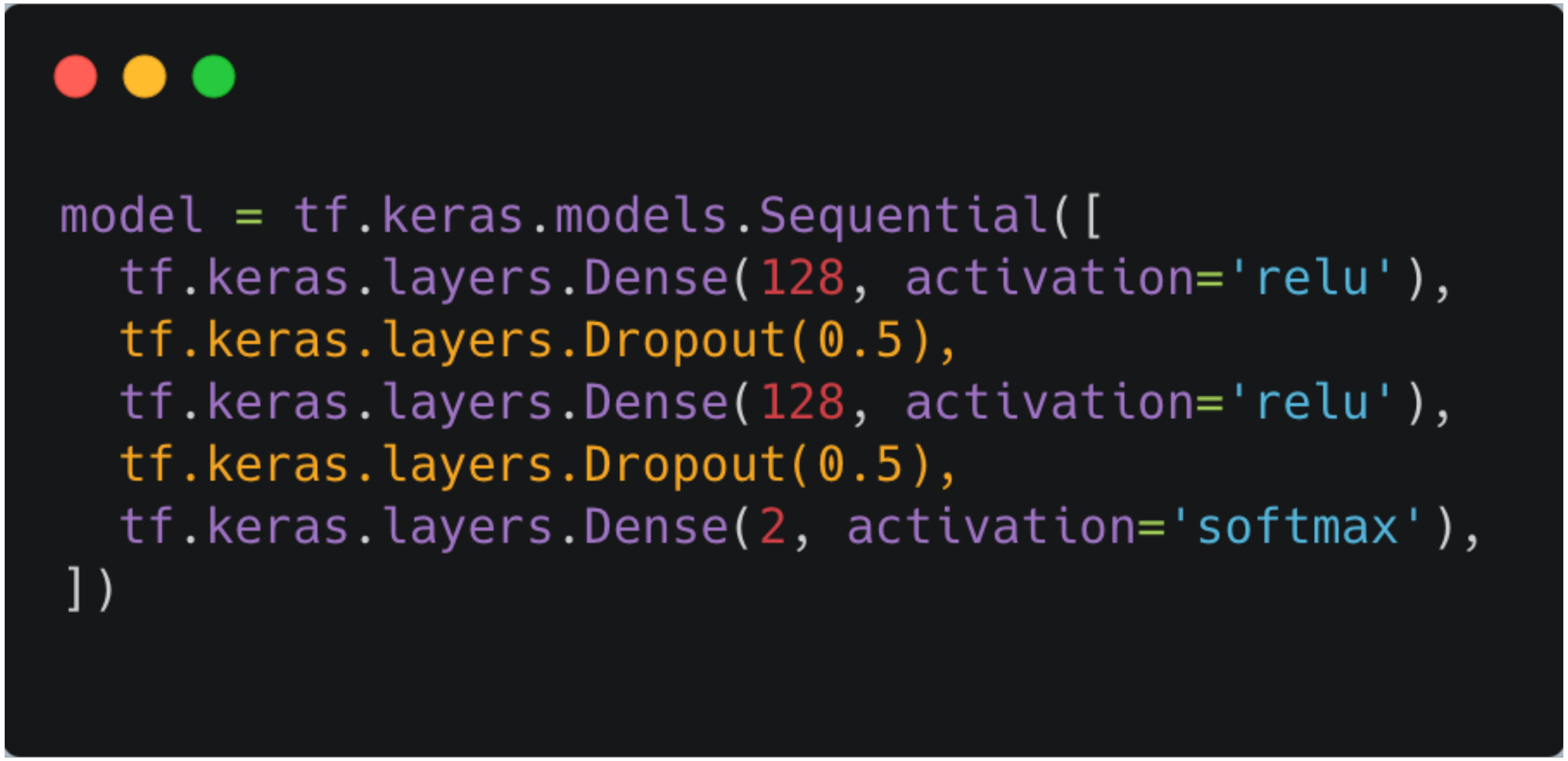




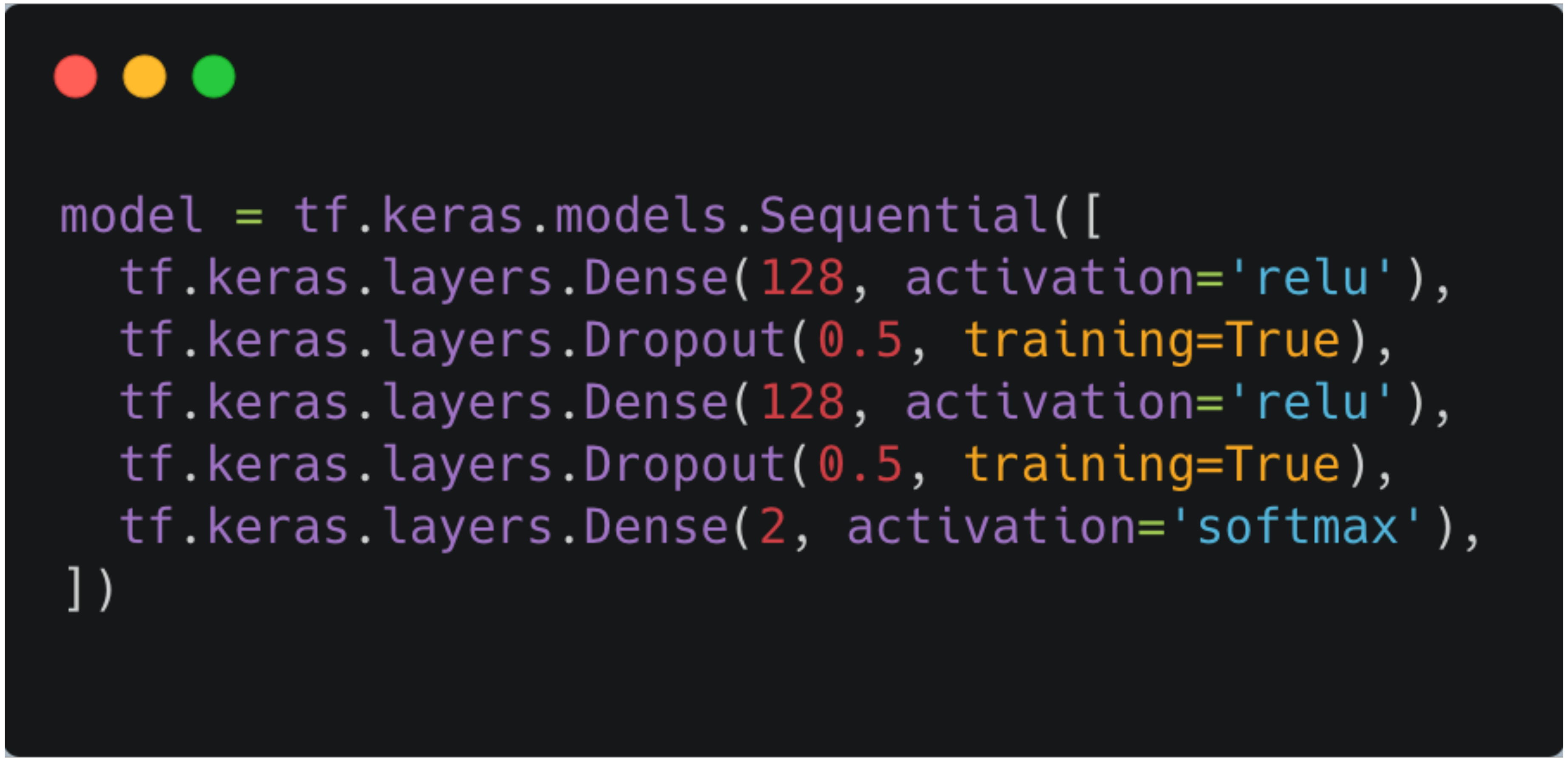
Cat?

Dog?

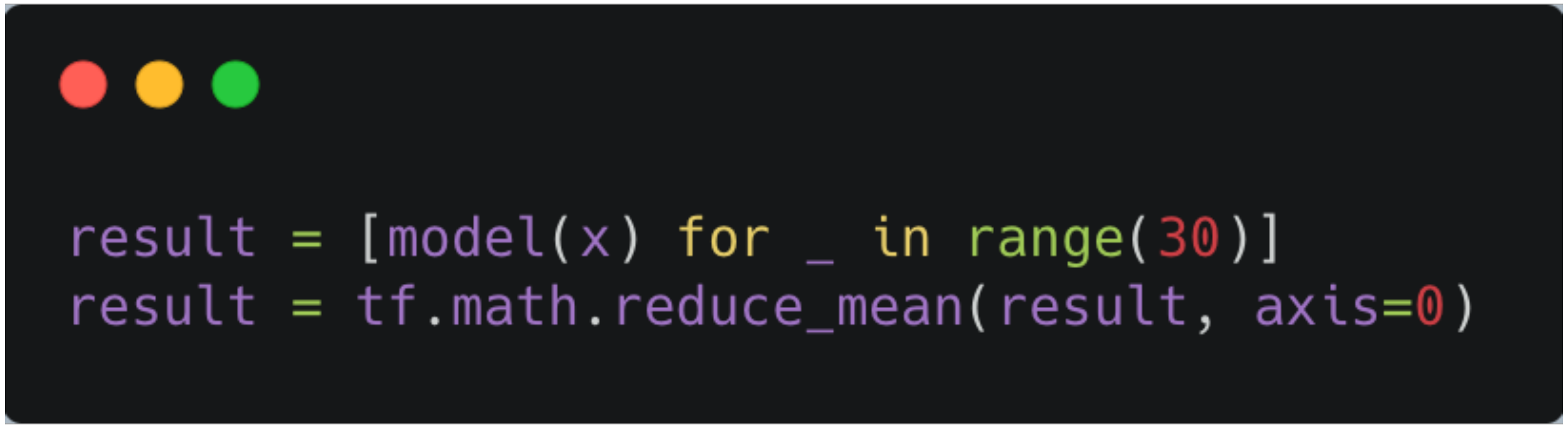
http://neuralnetworksanddeeplearning.com/chap6.html



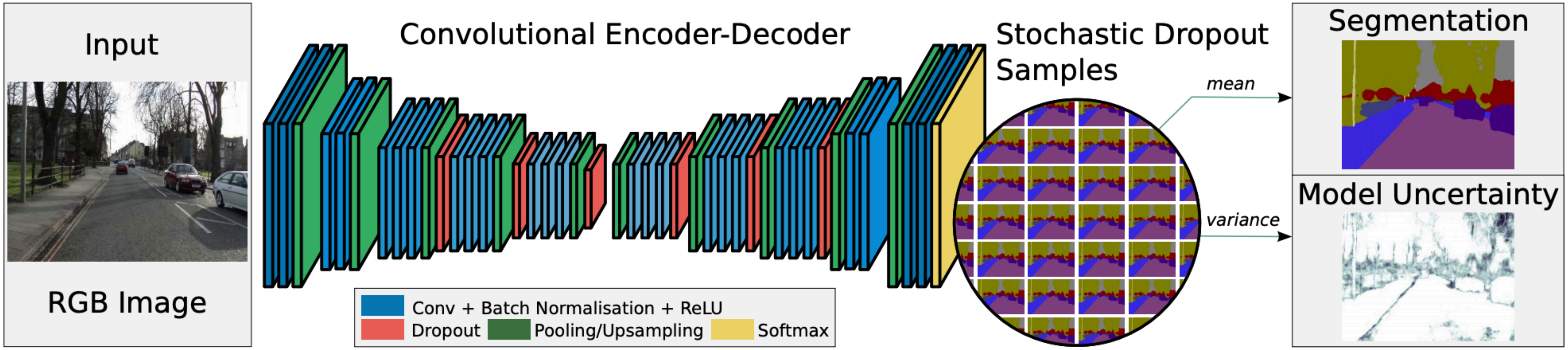
```
model = tf.keras.models.Sequential([
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.5),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.5),
  tf.keras.layers.Dense(2, activation='softmax'),
])
```



```
model = tf.keras.models.Sequential([
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.5, training=True),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.5, training=True),
  tf.keras.layers.Dense(2, activation='softmax'),
])
```

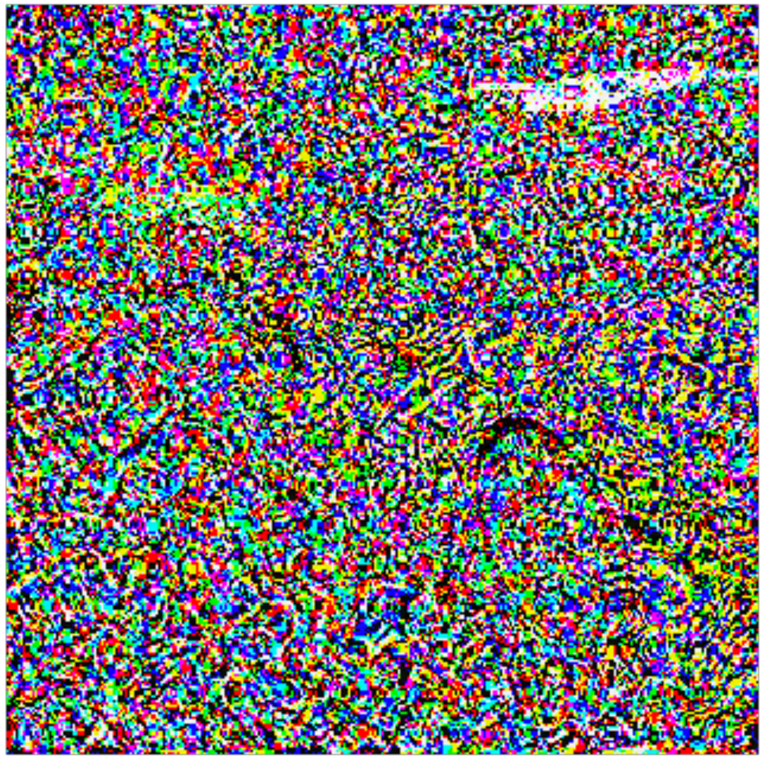


```
result = [model(x) for _ in range(30)]
result = tf.math.reduce_mean(result, axis=0)
```

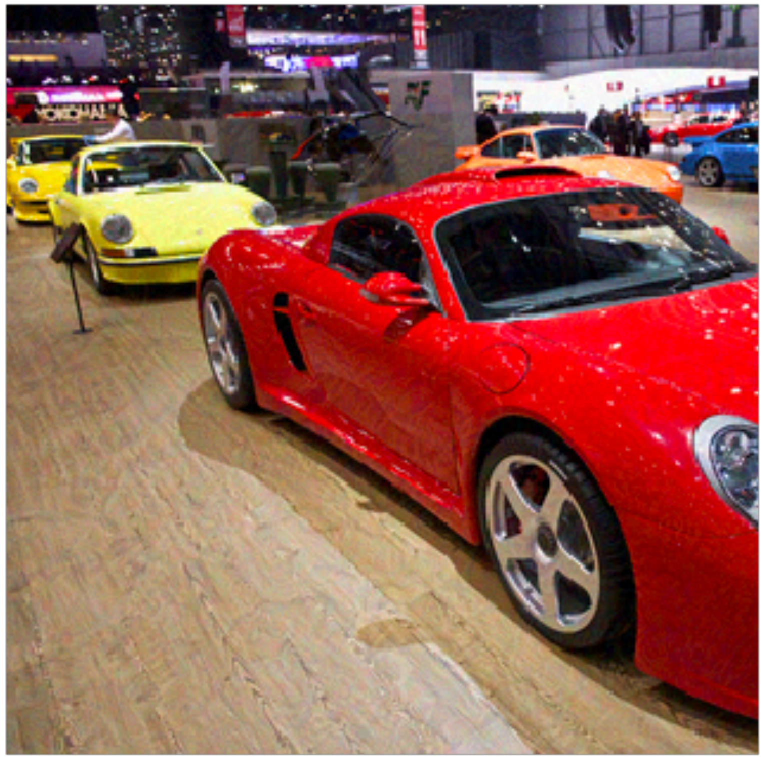

Input
RGB Image
Convolutional Encoder-Decoder
Conv + Batch Normalisation + ReLU
Dropout
Pooling/Upsampling
Softmax
Stochastic Dropout Samples
mean
variance
Segmentation
Model Uncertainty

Kendall, Alex, Vijay Badrinarayanan, and Roberto Cipolla. "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding." arXiv preprint arXiv:1511.02680 (2015).

Original Image

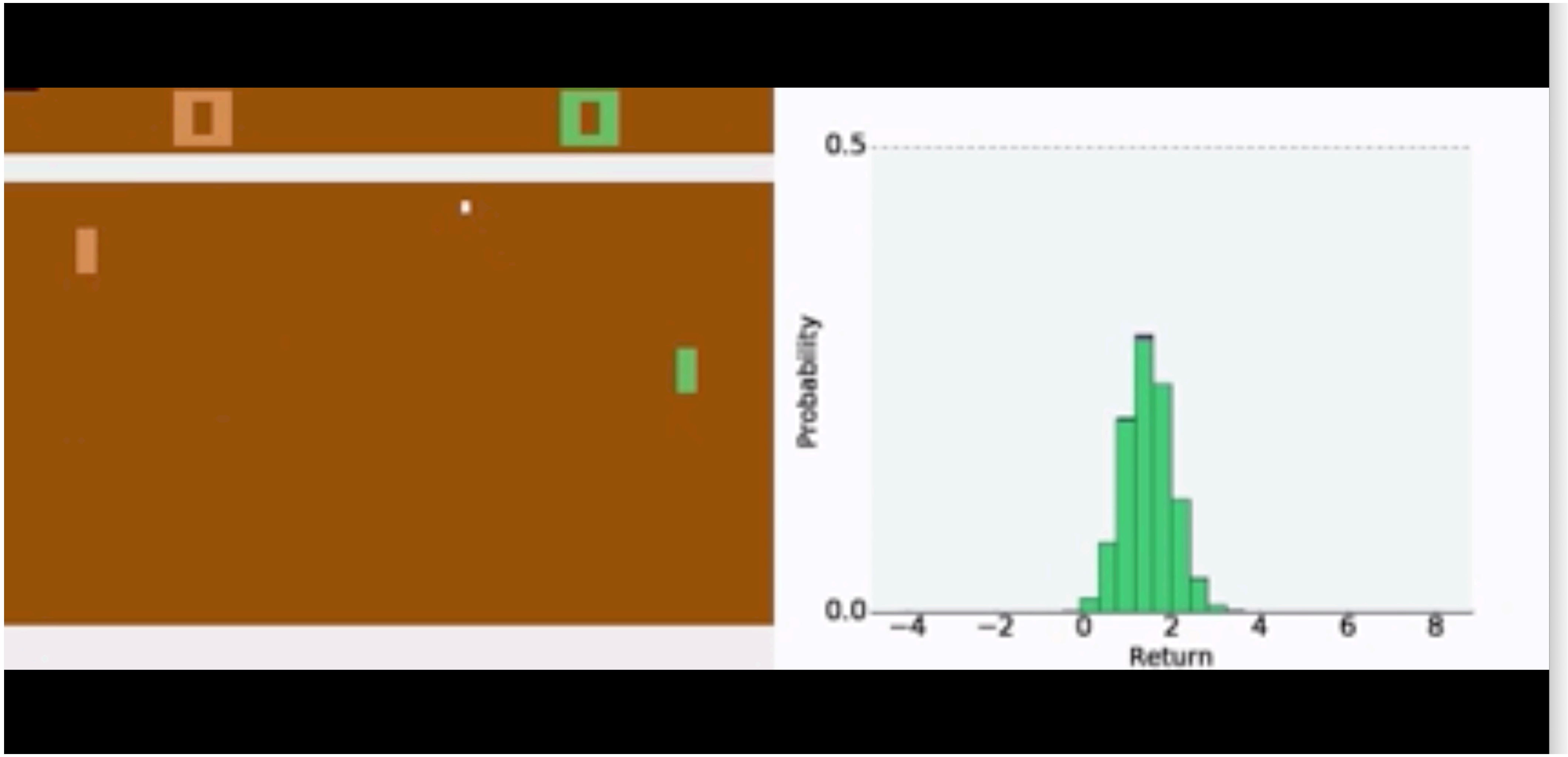


Noise

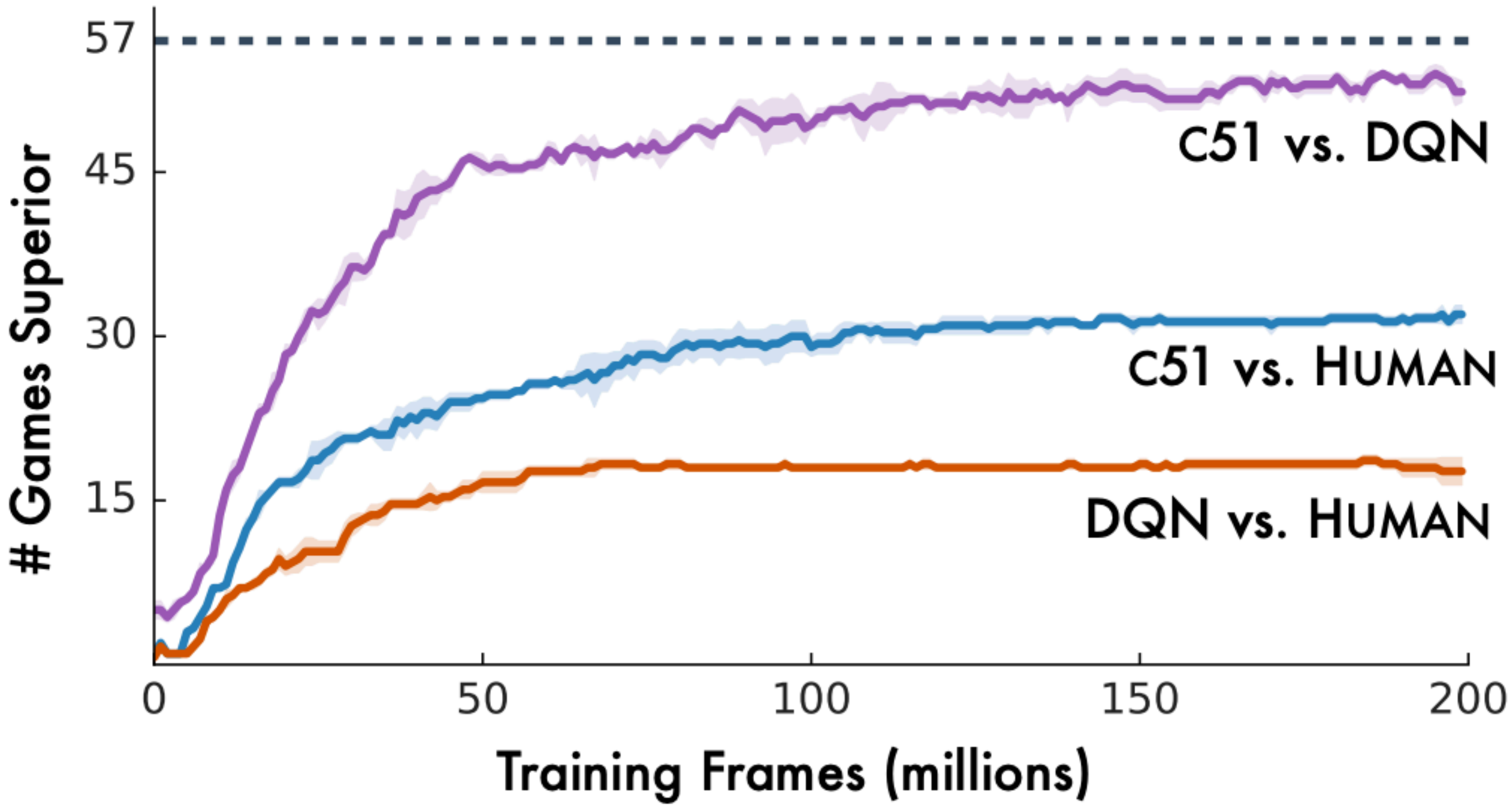


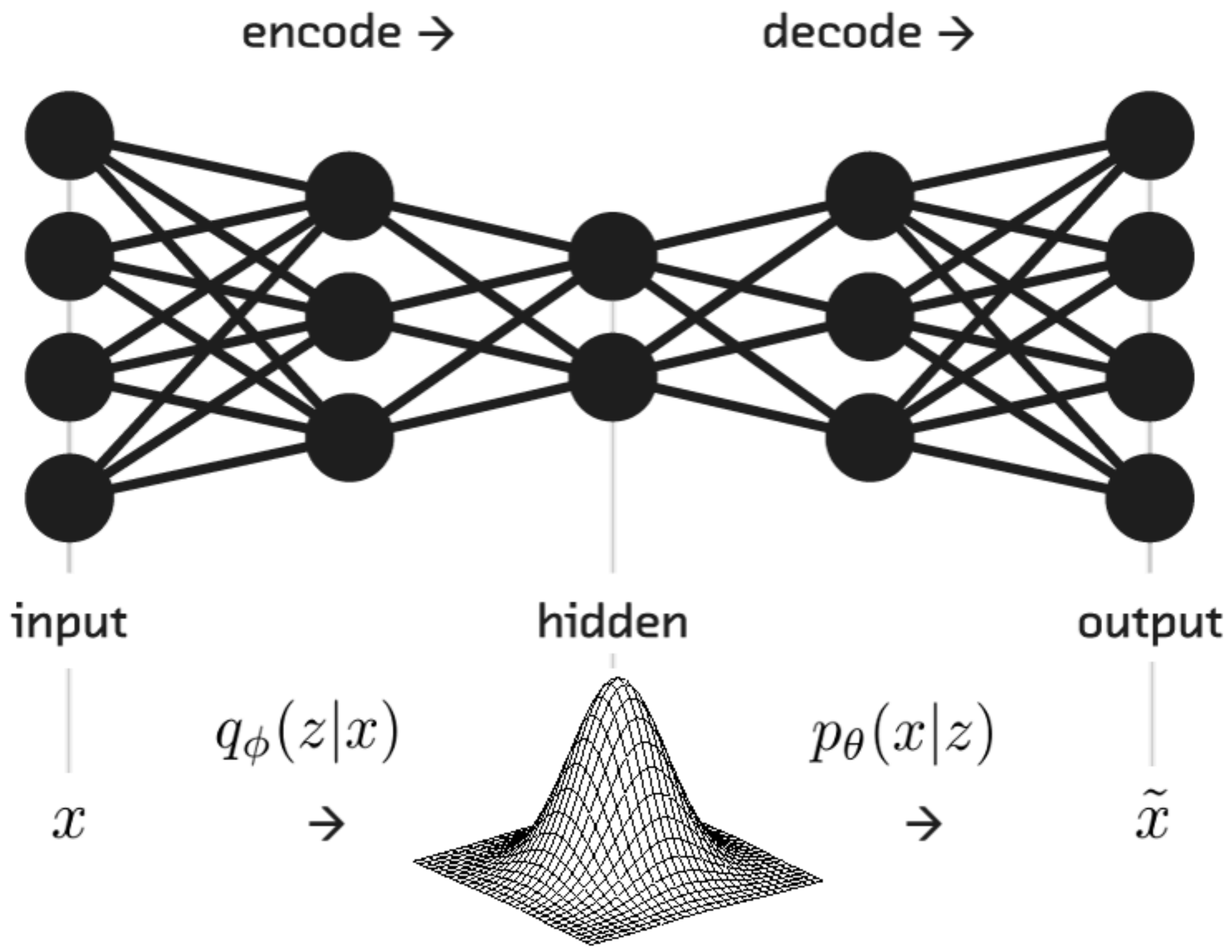
Advarsarial Image: ***Toaster?***

https://medium.com/@jrodthoughts/using-adversarial-attacks-to-make-your-deep-learning-model-look-stupid-24fb872f06fd

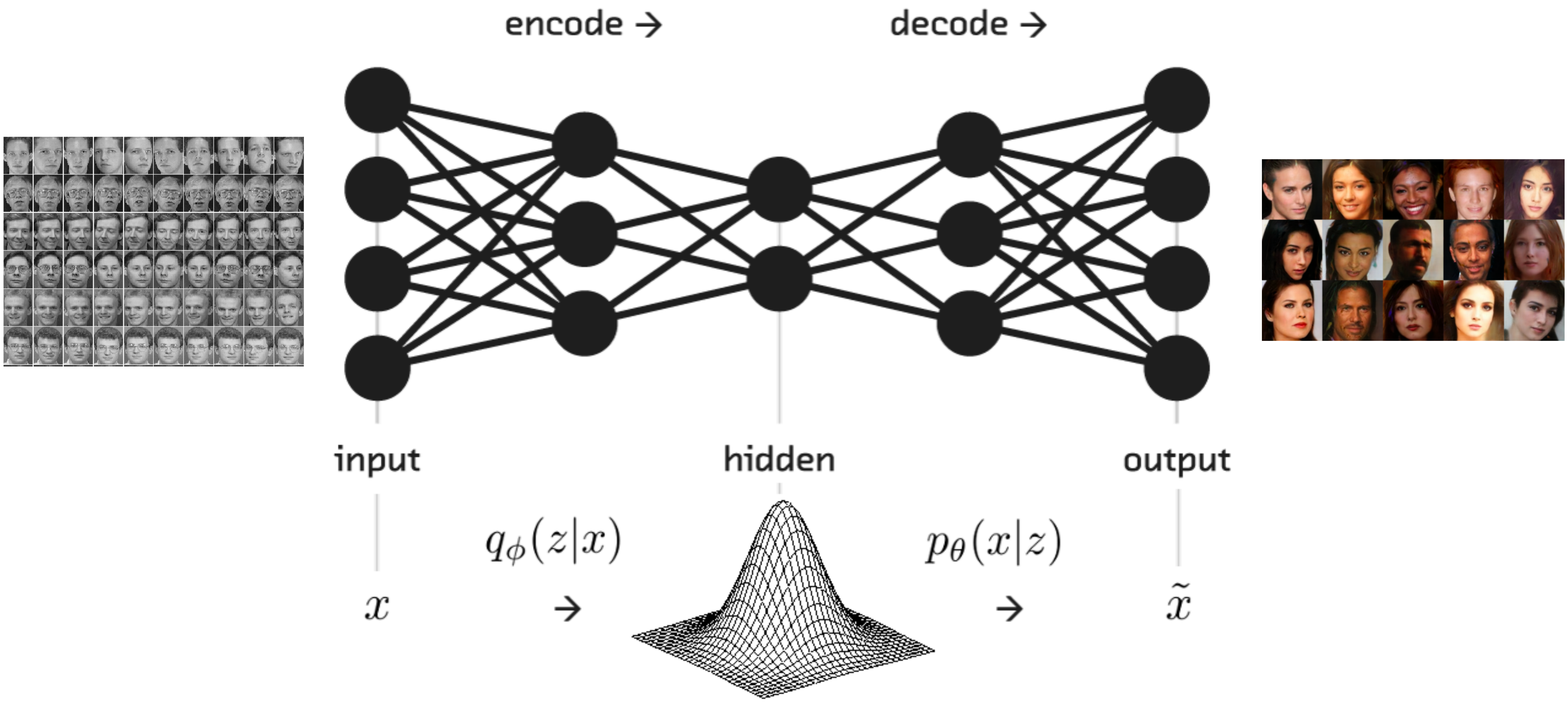


https://www.youtube.com/watch?v=vIz5P6s80qA&feature=youtu.be

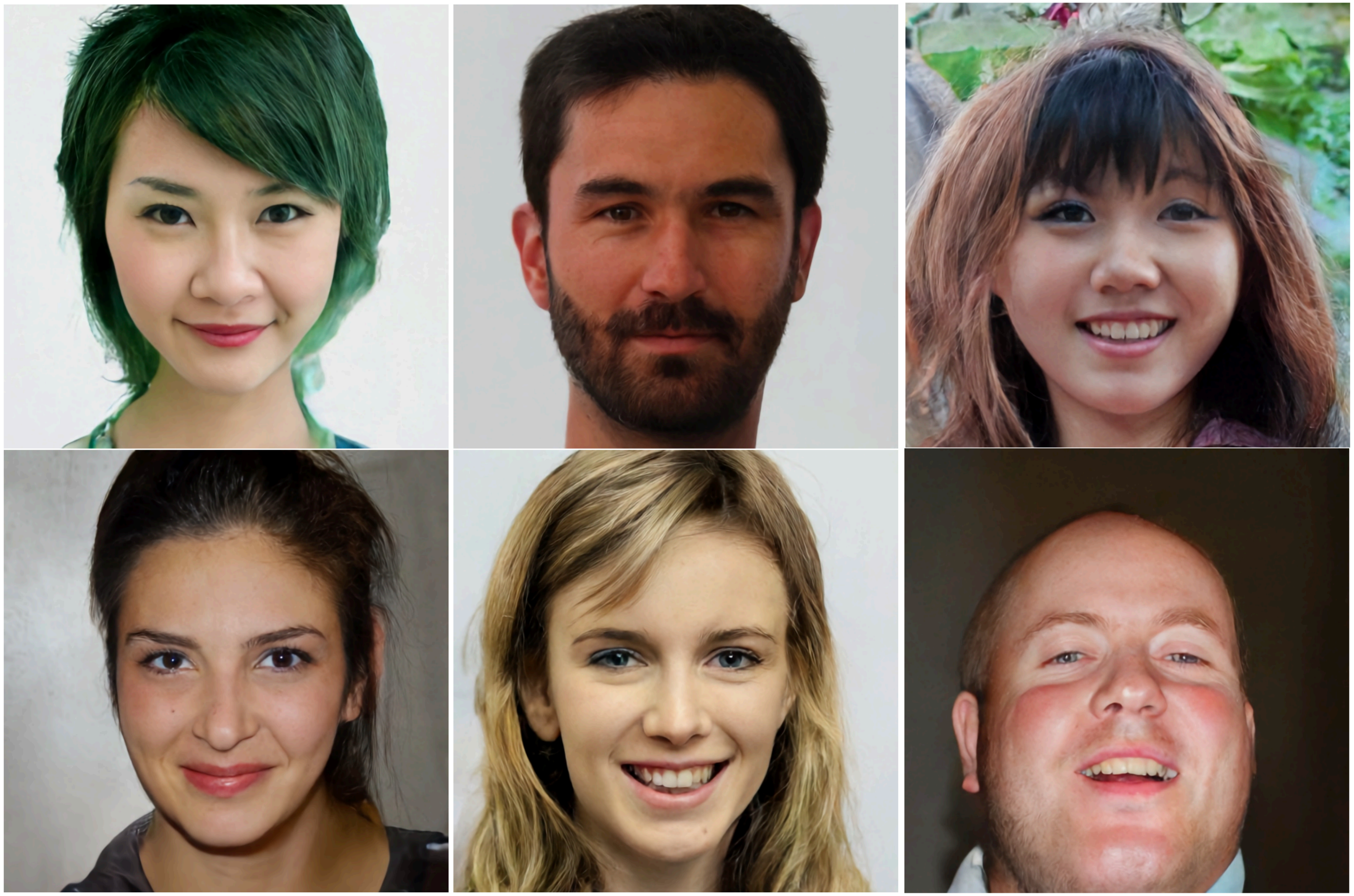

57
45
30
15
# Games Superior
0
50
100
150
200
Training Frames (millions)
C51 vs. DQN
C51 vs. HUMAN
DQN vs. HUMAN



https://towardsdatascience.com/what-the-heck-are-vae-gans-17b86023588a



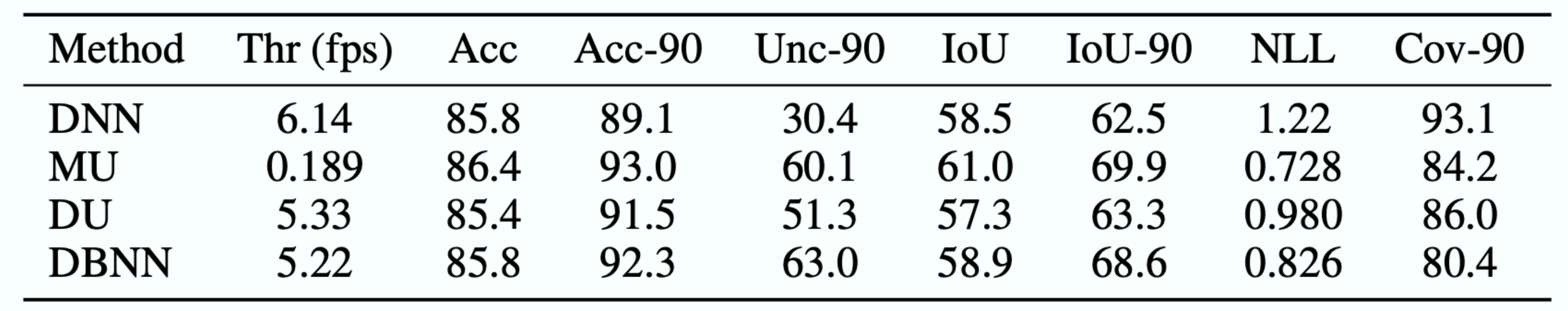
https://towardsdatascience.com/what-the-heck-are-vae-gans-17b86023588a



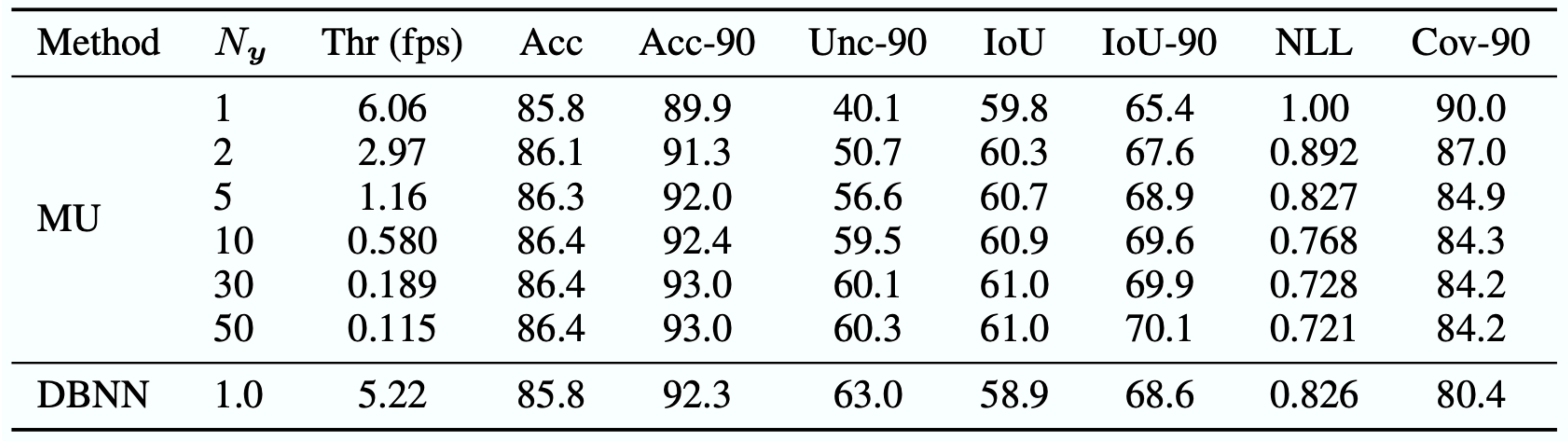
Razavi, Ali, Aaron van den Oord, and Oriol Vinyals. "Generating Diverse High-Fidelity Images with VQ-VAE-2." arXiv preprint arXiv:1906.00446 (2019).

- 결과만을 예측하는 것은 불충분하다. 결과±오차를 예측해야 한다.
- 오차를 예측하기 위해, 여러 모델의 앙상블을 고려해야 한다.
- 여러 모델을 앙상블하는 가장 쉬운 방법은 Dropout이다.

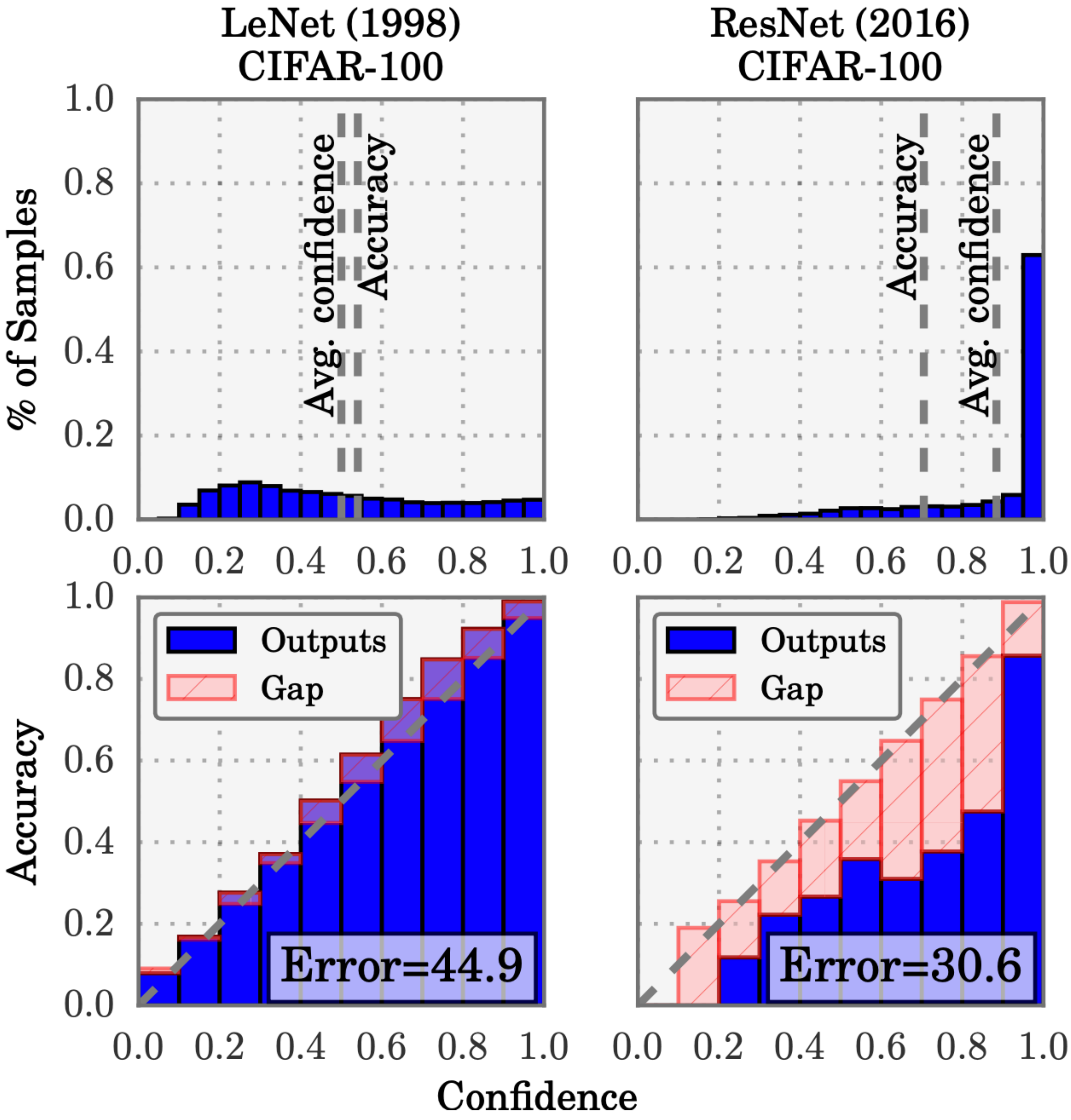
Appendix



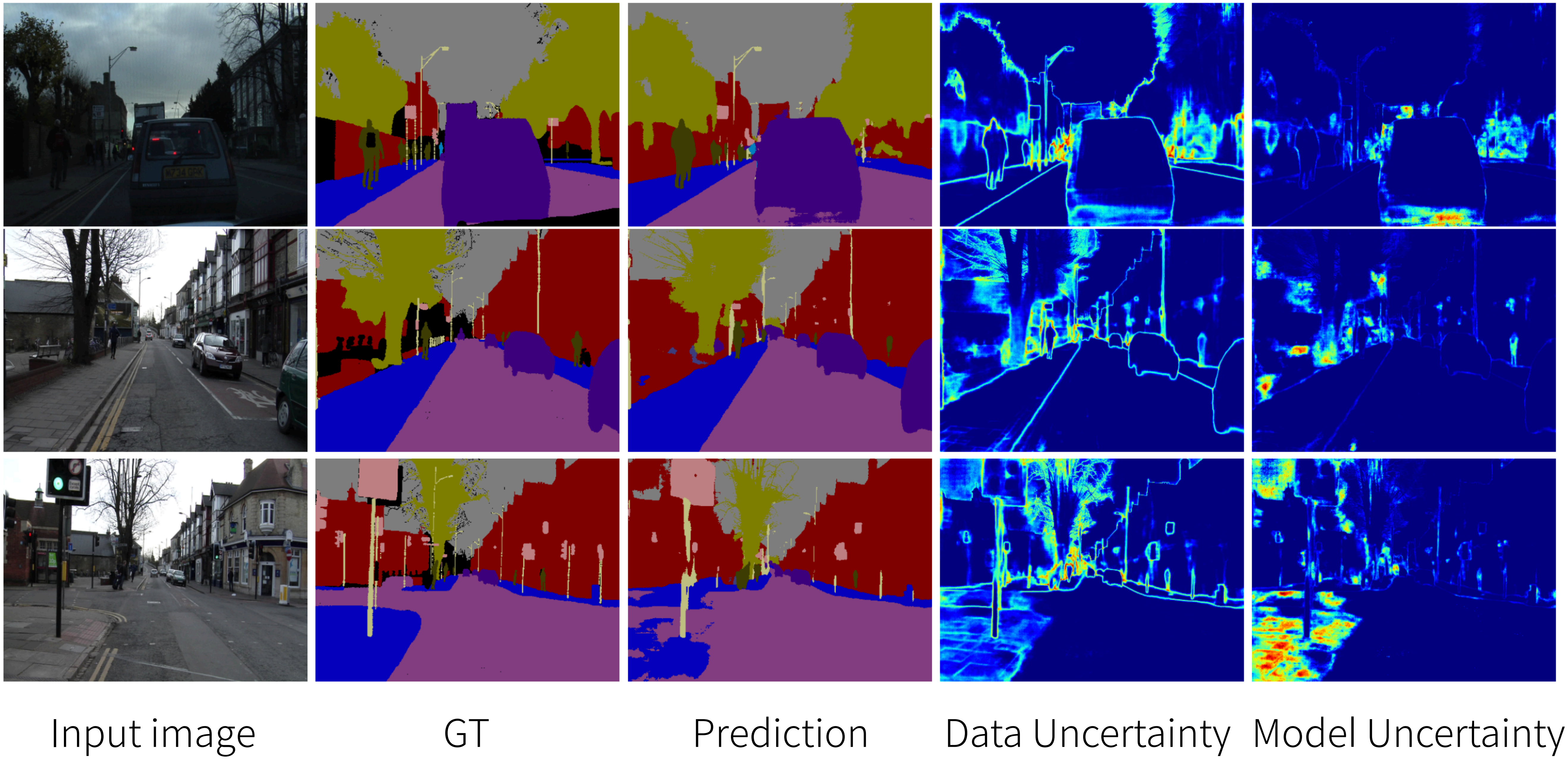
Method	Thr (fps)	Acc	Acc-90	Unc-90	IoU	IoU-90	NLL	Cov-90
DNN	6.14	85.8	89.1	30.4	58.5	62.5	1.22	93.1
MU	0.189	86.4	93.0	60.1	61.0	69.9	0.728	84.2
DU	5.33	85.4	91.5	51.3	57.3	63.3	0.980	86.0
DBNN	5.22	85.8	92.3	63.0	58.9	68.6	0.826	80.4



Method	$N_{\boldsymbol{y}}$	Thr (fps)	Acc	Acc-90	Unc-90	IoU	IoU-90	NLL	Cov-90
MU	1	6.06	85.8	89.9	40.1	59.8	65.4	1.00	90.0
	2	2.97	86.1	91.3	50.7	60.3	67.6	0.892	87.0
	5	1.16	86.3	92.0	56.6	60.7	68.9	0.827	84.9
	10	0.580	86.4	92.4	59.5	60.9	69.6	0.768	84.3
	30	0.189	86.4	93.0	60.1	61.0	69.9	0.728	84.2
	50	0.115	86.4	93.0	60.3	61.0	70.1	0.721	84.2
DBNN	1.0	5.22	85.8	92.3	63.0	58.9	68.6	0.826	80.4

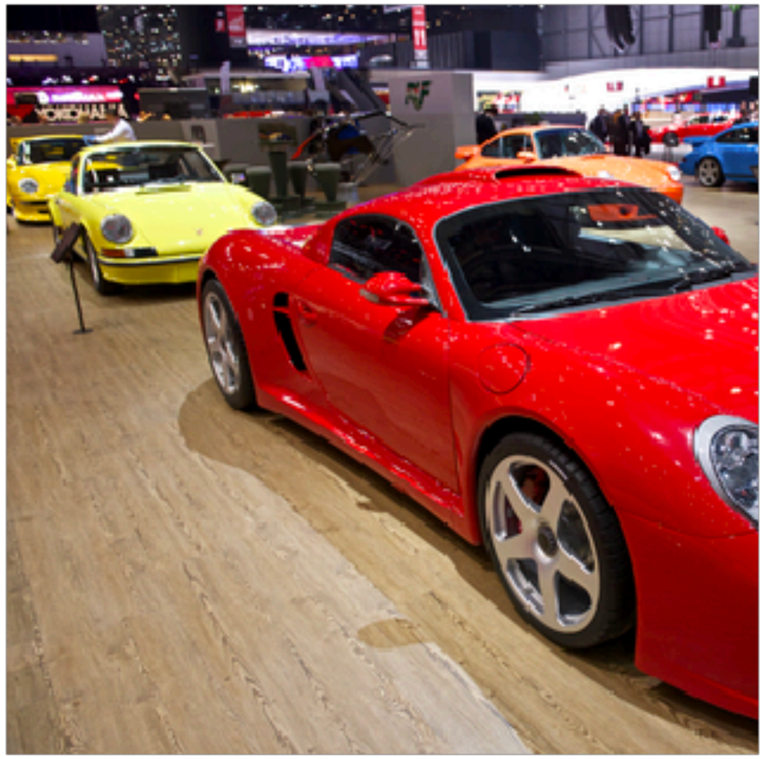


Guo, Chuan, et al. "On calibration of modern neural networks." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.

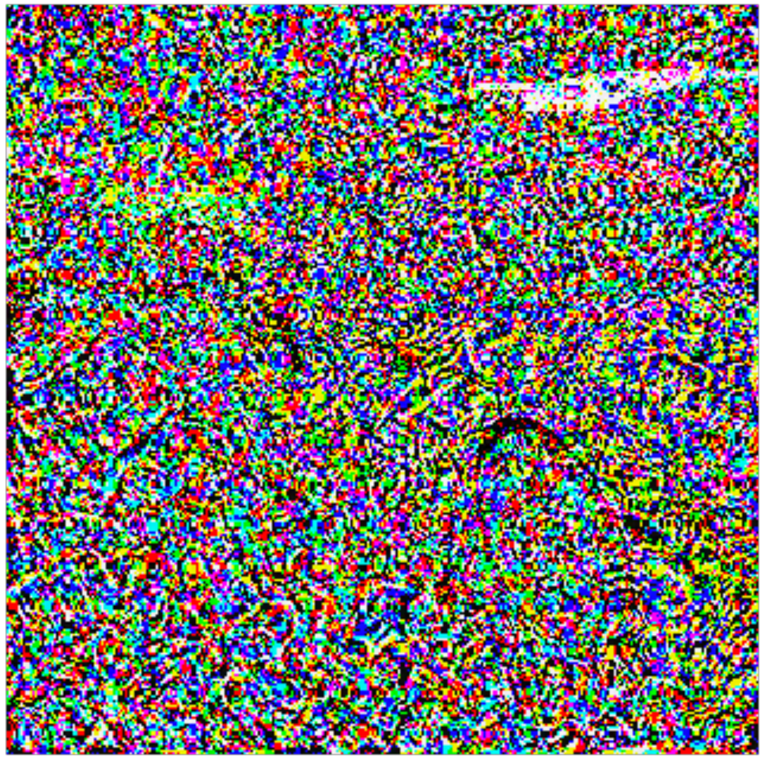


Input image GT Prediction Data Uncertainty Model Uncertainty

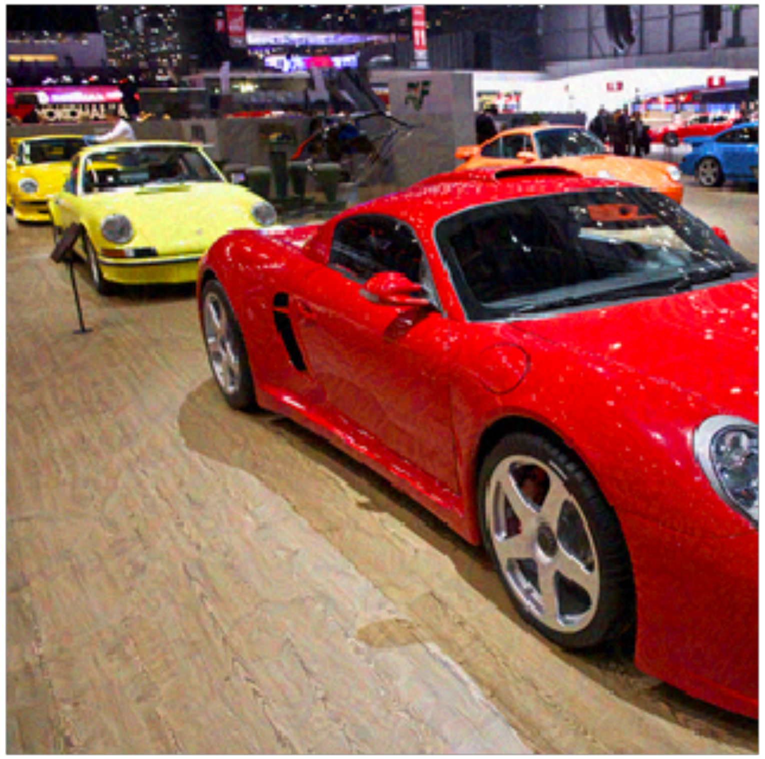
Kendall, Alex, and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?." Advances in neural information processing systems. 2017.



Original Image

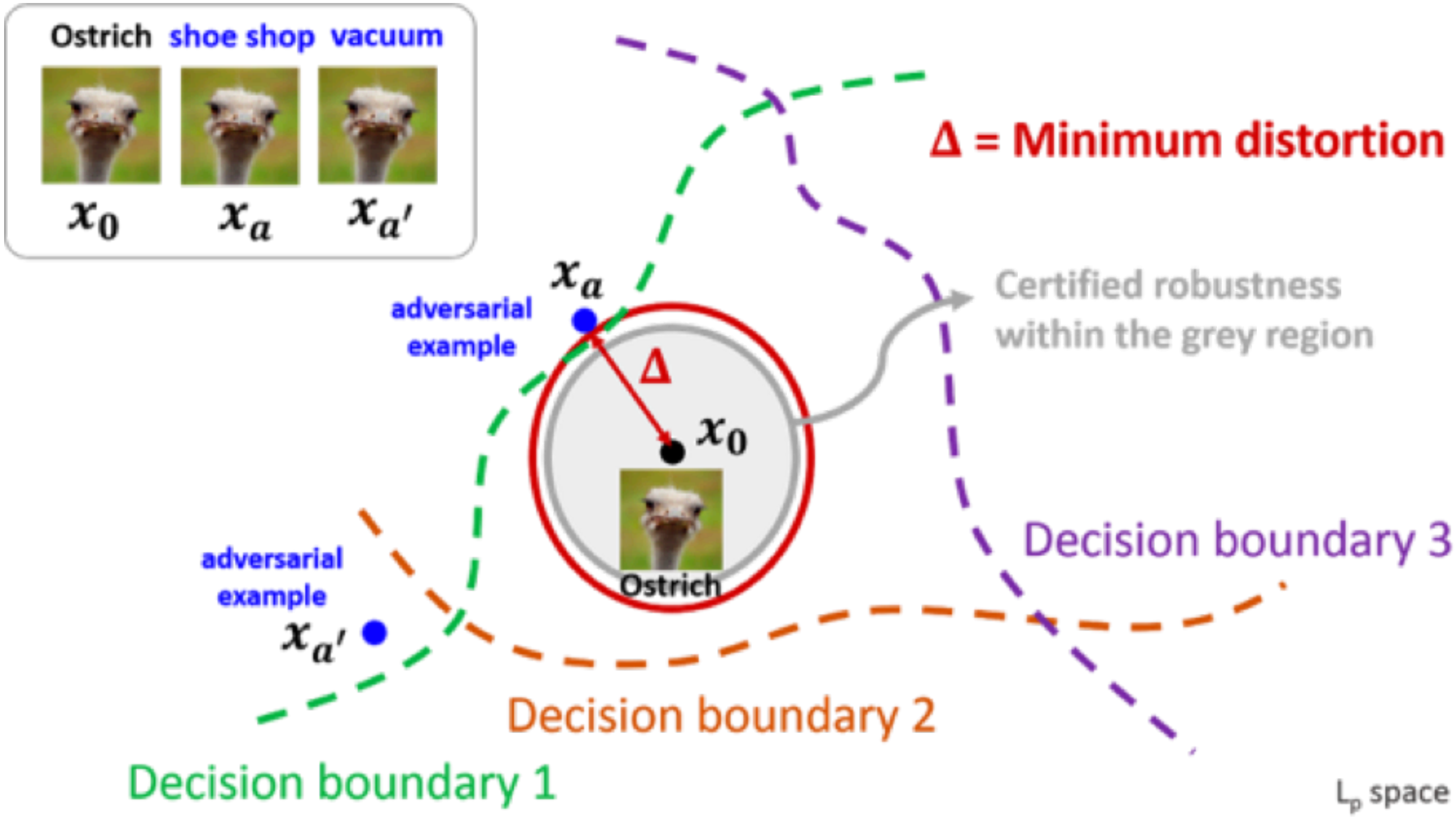


Noise

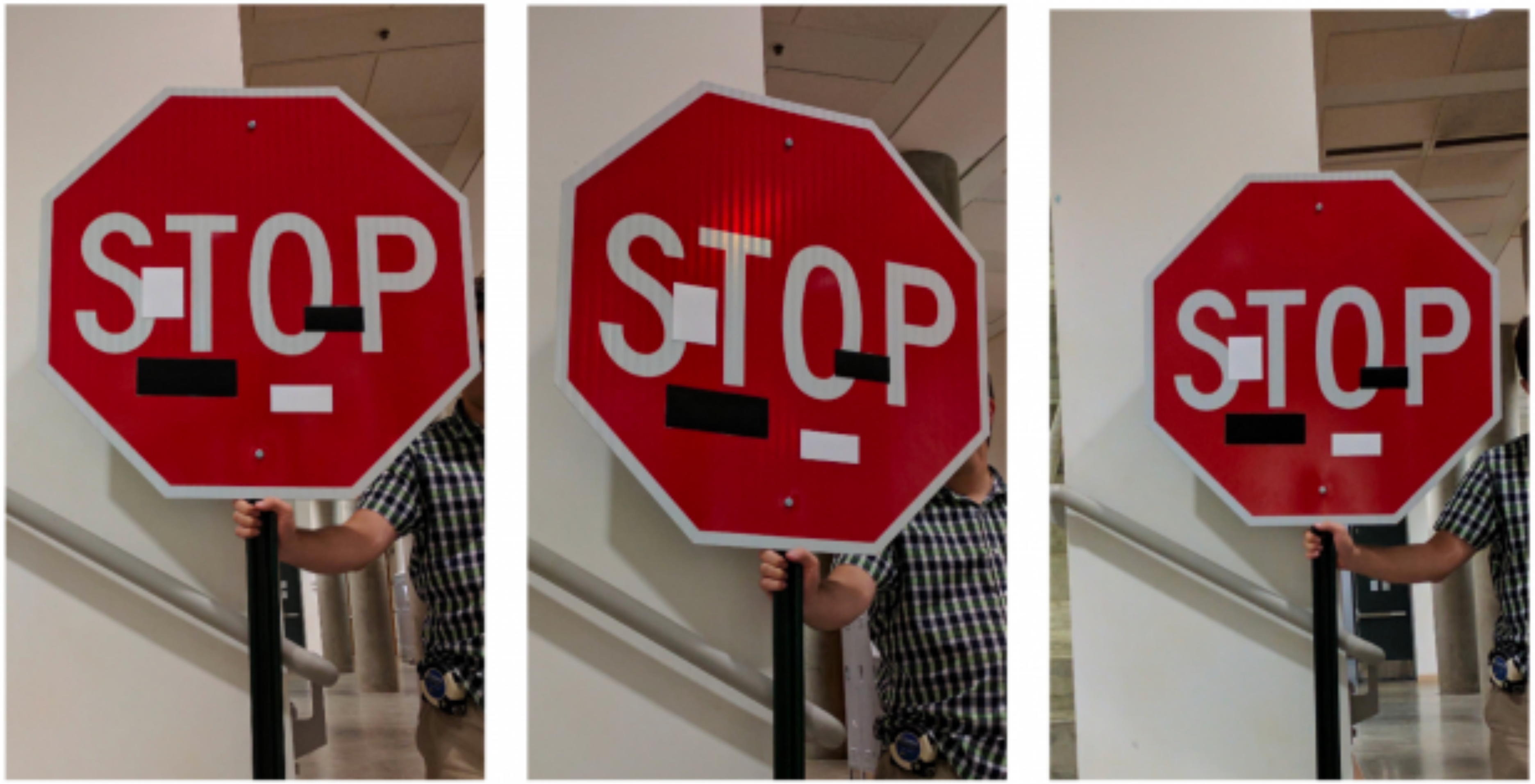


Advarsarial Image: ***Toaster?***

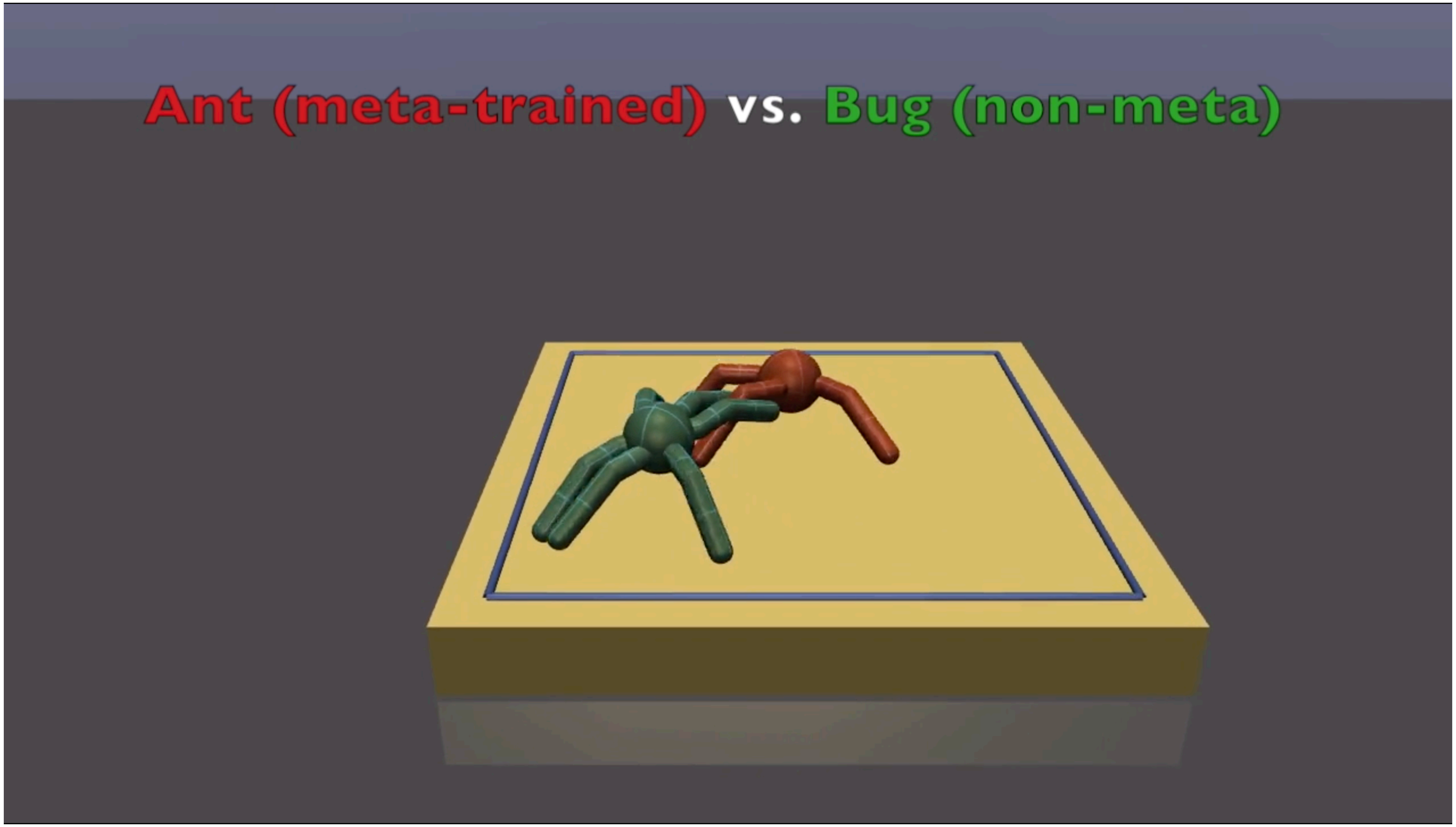
https://medium.com/@jrodthoughts/using-adversarial-attacks-to-make-your-deep-learning-model-look-stupid-24fb872f06fd



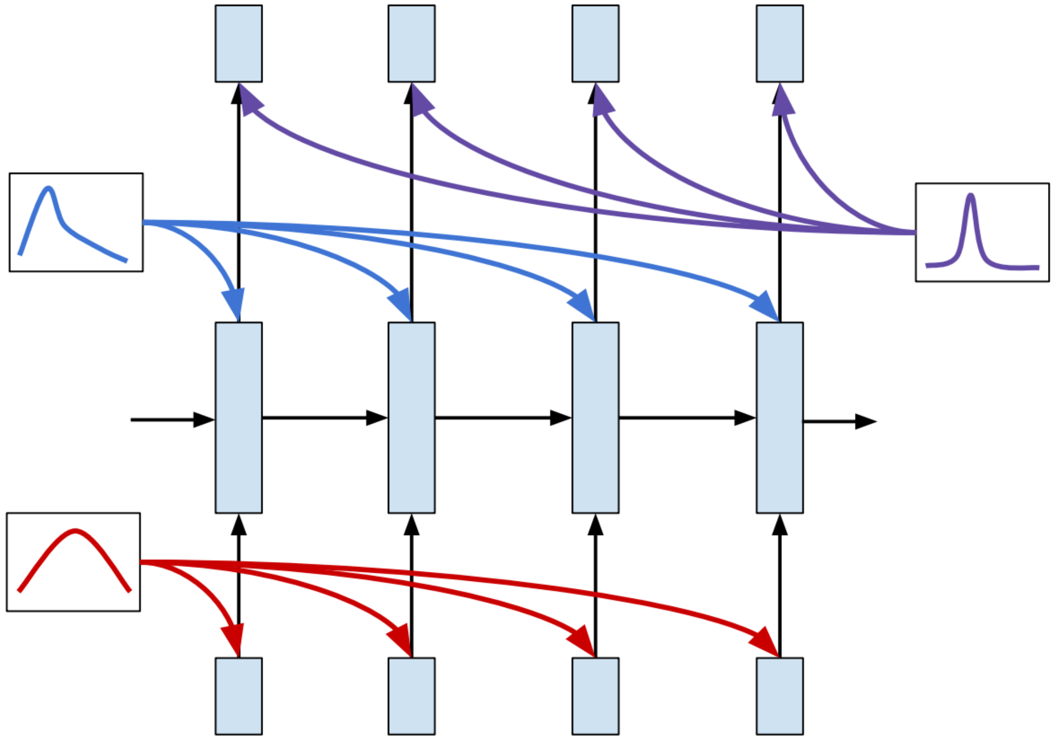
https://www.ibm.com/blogs/research/2018/05/clever-adversarial-attack/



https://medium.com/self-driving-cars/adversarial-traffic-signs-fd16b7171906



https://d4mucfpksywv.cloudfront.net/multi_agent/adaptation-single-movie+(2).mp4



Baseline: a white plate with a pizza on it
BBB: a small white dog eating a piece of pizza

Baseline: a small boat in a large body of water
BBB: a boat traveling down a river next to a bridge

Fortunato, Meire, Charles Blundell, and Oriol Vinyals. "Bayesian recurrent neural networks." arXiv preprint arXiv:1704.02798 (2017).