Why Does Deep Learning Work?

Aaron Mishkin

UBC MLRG 2019W1

Why Does Deep Learning Work?

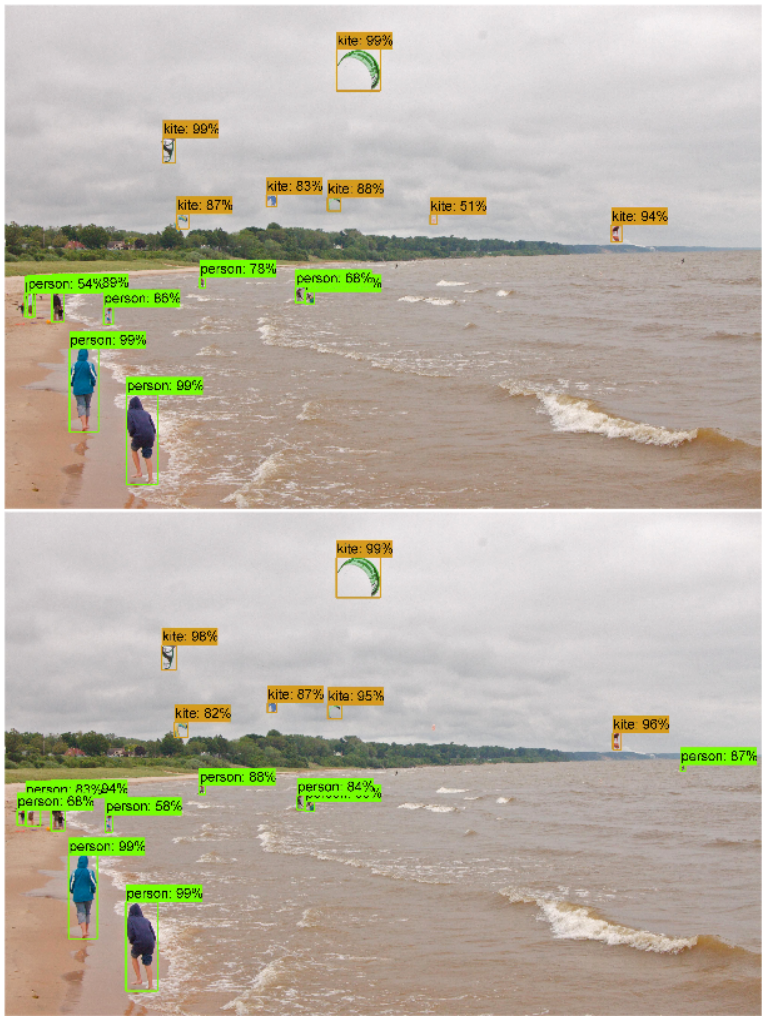
Common refrains from deep learning:

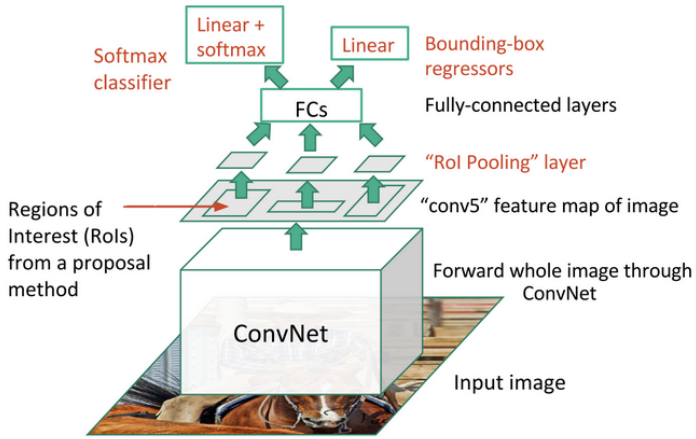
- "Always make your neural network **as big as possible**!"
- "Neural networks **generalize** because they're trained with **stochastic gradient descent (SGD)**."
- "**Sharp minima** are bad and **shallow** minima are good."
- "**SGD** finds **flat** local minima."

Where do these ideas come from?

Deep Learning Works!

Deep Learning Works: Object Localization





Object localization with Fast R-CNNs [1].

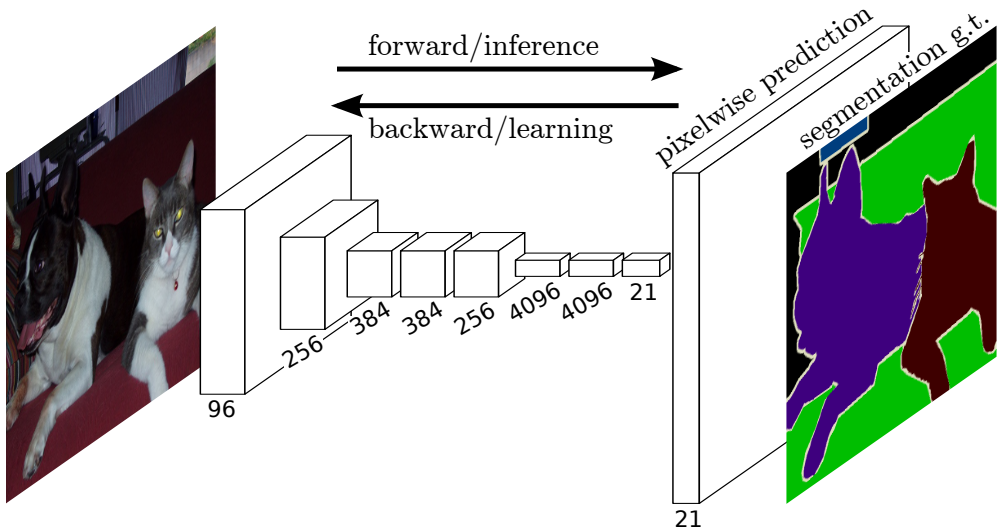
---

https://arxiv.org/abs/1707.07012

https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9

Deep Learning Works: Image Segmentation

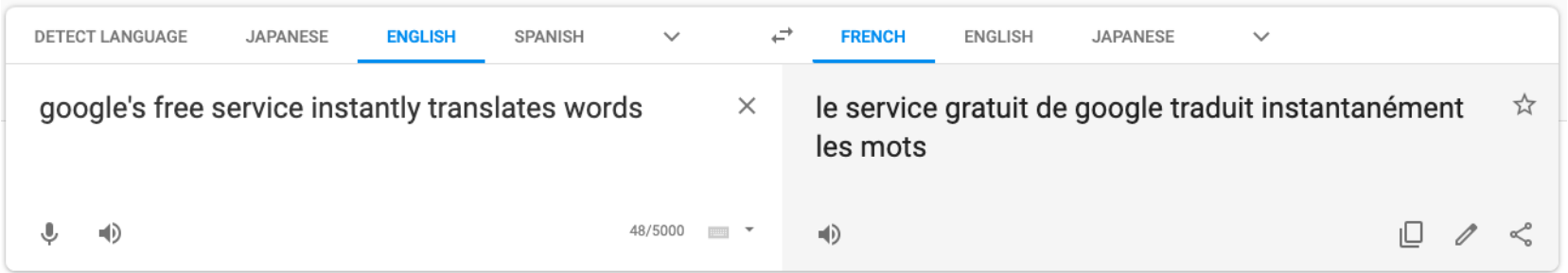
Image segmentation using fully convolutional networks [3].



https://arxiv.org/abs/1411.4038

https://arxiv.org/abs/1802.02611

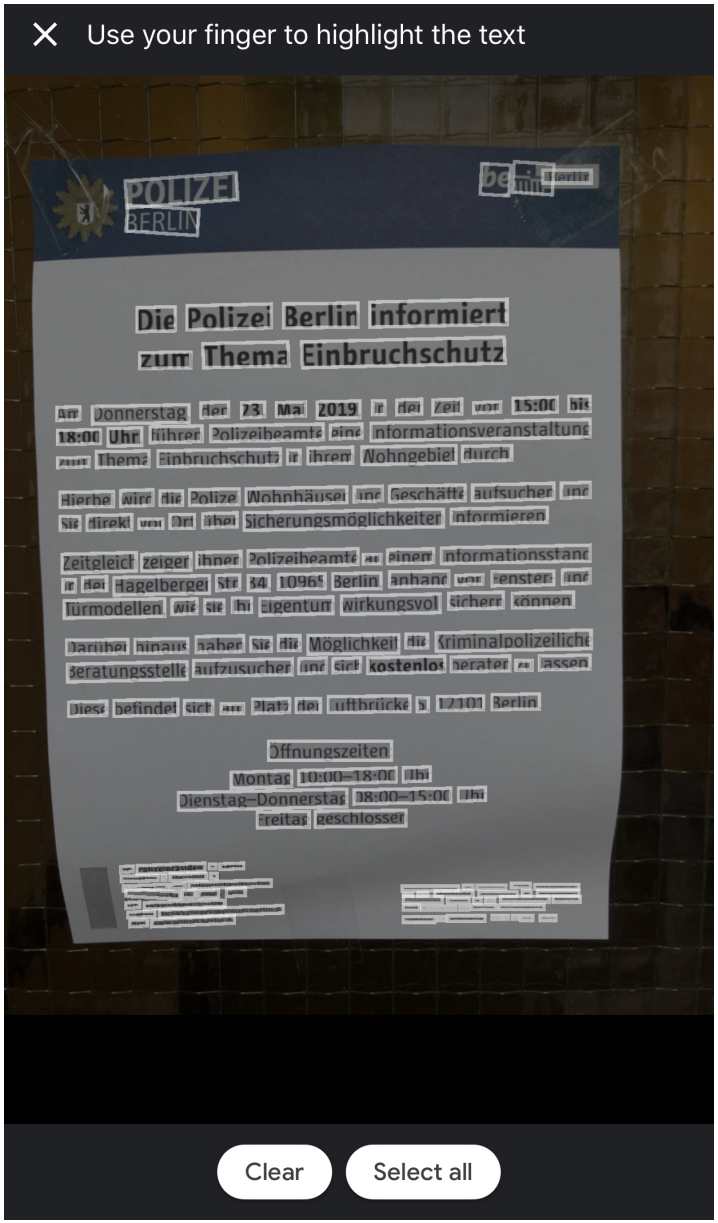
Deep Learning Works: Machine Translation (1)



Google's Neural Machine Translation System:

- consists of a **deep LSTM network** with 8 encoder and 8 decoder layers using attention and residual connections.
- reduced translation errors "by an average of **60%** compared to Google's phrase-based" system.

Deep Learning Works: Machine Translation (2)



Berlin POLIZEI BERLIN The Berlin police informs about burglary protection On Thursday, 23rd May 2019, between 3:00 pm and 6:00 pm, police officers hold an information event on burglary protection in their residential area. In the process, the police will visit residential buildings and shops and inform you directly about security options. At the same time, police officers at an information stand in the Hagelberger Str. 34, 10965 Berlin show them, with the help of window and door models, how they can effectively secure their property...

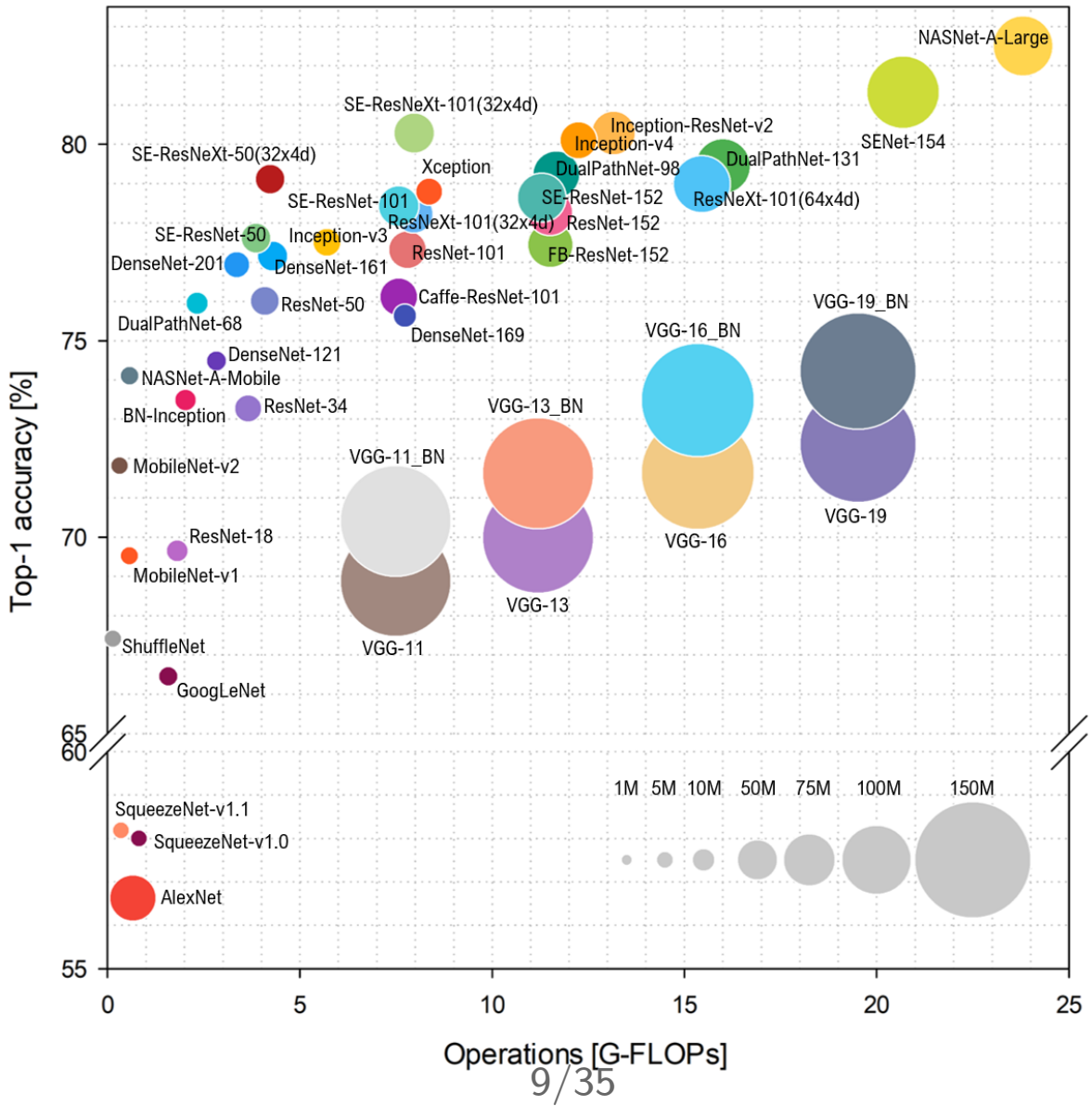
Deep Learning Works: Generative Models

StyleGAN: image generatation with hierarchical style transfer [2].

https://arxiv.org/abs/1812.04948

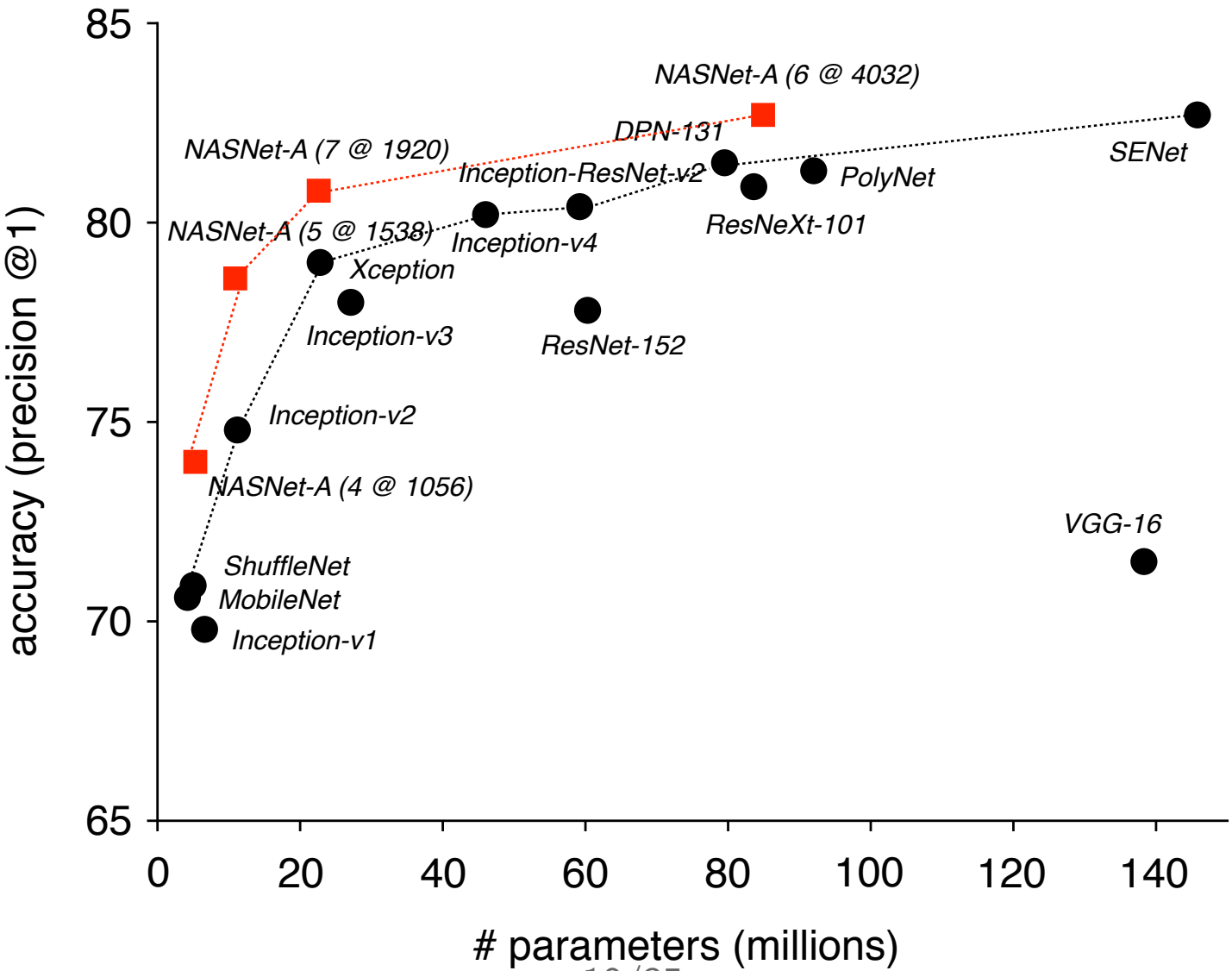
Deep Learning Works: Model Sizes (1)

Accuracy on ImageNet (2012 ILSVRC)



Deep Learning Works: Model Sizes (2)

Accuracy on ImageNet (2012 ILSVRC): Exponentially more parameters are needed to improve accuracy.

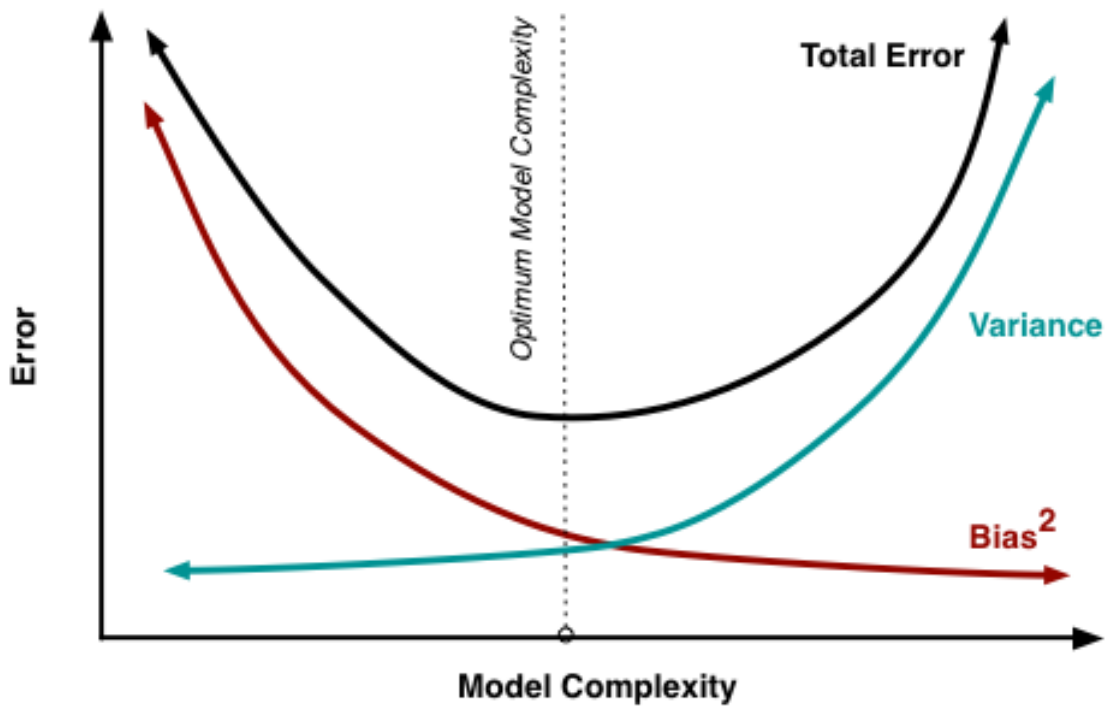


Deep Learning Works: Bias-Variance

But what about the **bias-variance** trade-off?

Let $y = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$,

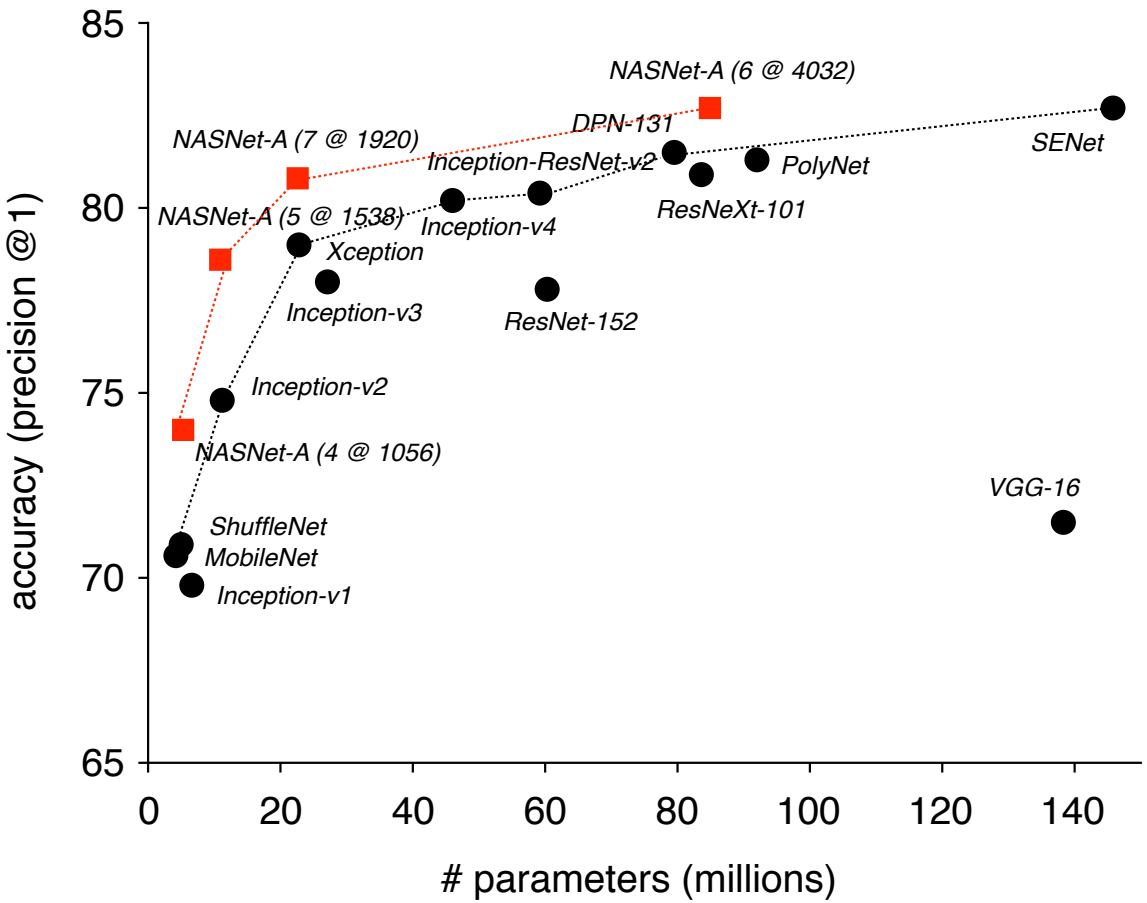
$$\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \textbf{Bias}\left[\hat{f}(x)\right]^2 + \textbf{Var}\left[\hat{f}(x)\right] + \sigma^2$$



Bias-Variance: Deep Models

We expect bigger architectures to:

- have **lower bias** because have they more parameters,
- but **higher variance** across different training sets.

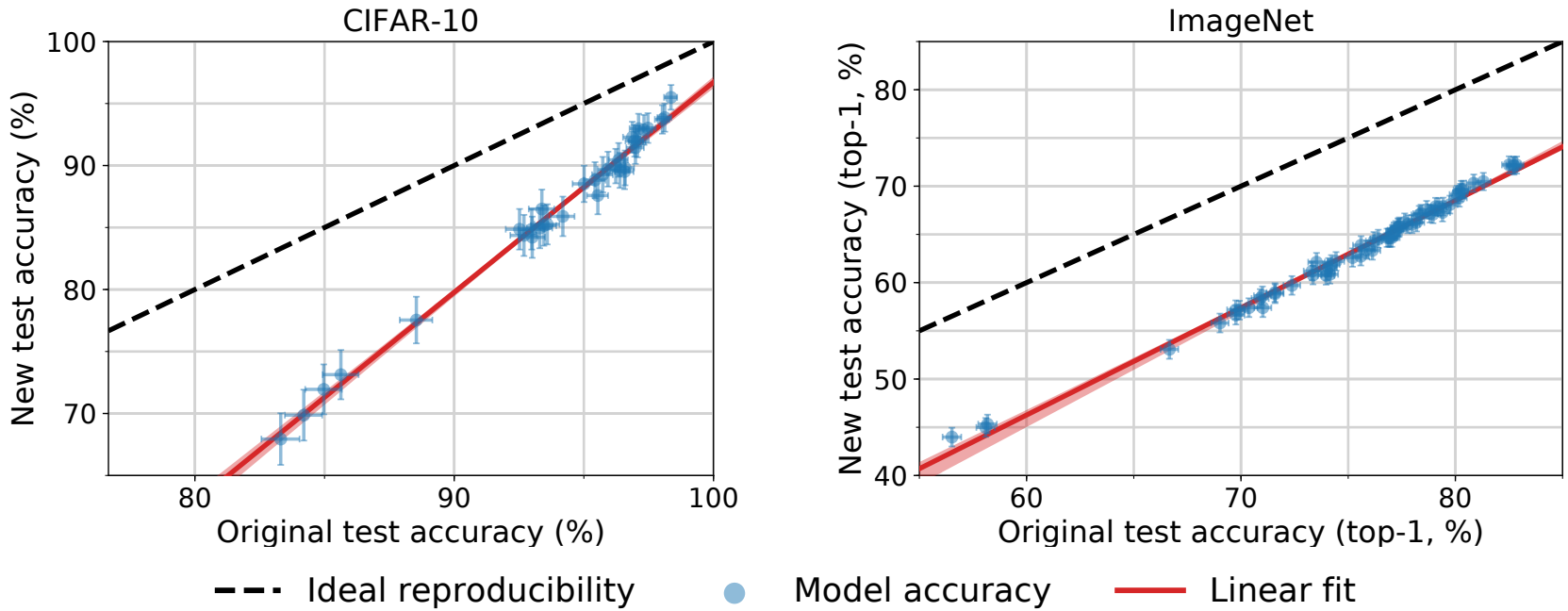


Shouldn't we see this in action with deep models?

https://arxiv.org/abs/1707.07012

Bias-Variance: Optimization Bias

Maybe we're **overfitting** architectures to the test set!



New Test Sets for CIFAR-10 and ImageNet [4]:

- "the relative order of models is almost **exactly preserved**"
- "there are **no diminishing returns** in accuracy"

https://arxiv.org/abs/1902.10811

Why do bigger neural networks lead to better accurracy?

The issue is how we think about model "capacity".

Perceptron: An Instructive Example

Learning Theory: A Brief Introduction

Statistical learning theory tries to develop guarantees for the performance of machine learning models.

- Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be a **training set** of input-output pairs.
  - $\mathcal{D}$ is formed by sampling $(x, y) \sim p(x, y)$ n times.
- Let $\mathcal{H}$ be a **hypothesis class**.
  - $\mathcal{H}$ is a fixed set of prediction functions $f(x) = \hat{y}$ that we pre-select.
  - $\mathcal{H}$ could be SVMs with RBF kernels, one-layer neural networks, etc.
- A **learning algorithm** takes $\mathcal{D}$ as input and returns $\hat{f} \in \mathcal{H}$.

What does it mean to **generalize** in this framework?

Learning Theory: Risk and ERM

Let $\mathcal{L}$ be a loss function. We care about the **risk**,

$$R(f) = \mathbb{E}_{p(x,y)} \left[\mathcal{L}(f(x), y) \right].$$

We don't know $p(x, y)$, but we do have the training set $\mathcal{D}$. The **empirical risk** is simply the loss on $\mathcal{D}$,

$$R_{\mathcal{D}}(f) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i).$$

Empirical risk minimization (ERM) is the learning algorithm

$$\hat{f} = \min_{f \in \mathcal{H}} R_{\mathcal{D}}(f) = \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i).$$

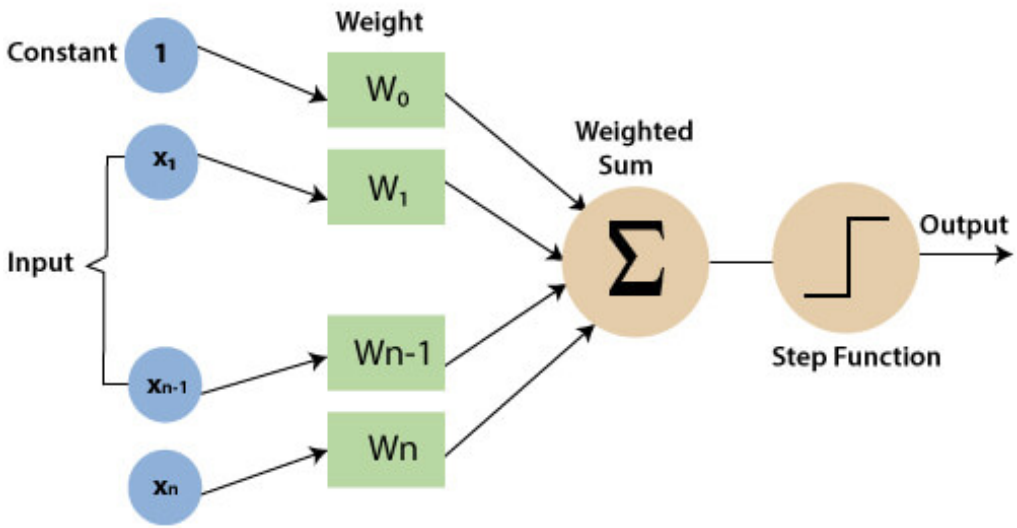
This is simple – choose $\hat{f}$ to minimize the training loss.

Perceptron: Definition

Perceptron is an early linear model for binary classification.

- Let $x \in \mathbb{R}^d$ and $y \in \{-1, 1\}$.
- $\mathcal{H}$ is the set of hyper-planes defined by $w \in \mathbb{R}^d$.
- Given weight vector w, $f_w(x) = \text{sign}(\langle w, x \rangle)$.

Perceptron is a neural network with one unit and sign activation.



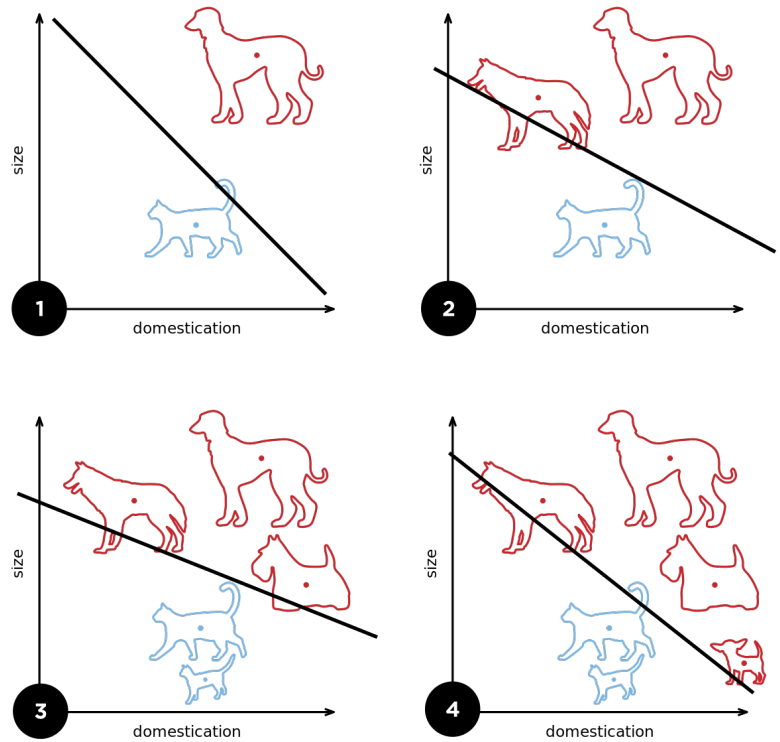
https://www.javatpoint.com/pytorch-perceptron

Perceptron: Learning Algorithm

Perceptron works by iteratively correcting its mistakes.

Algorithm 1 Perceptron Algorithm

$w_0 \leftarrow 0$
for $t = 0 \dots N-1$ **do**
  select $(x_t, y_t) \in \mathcal{D}$
  if $\text{sign}(\langle w_t, x_t \rangle) \neq y_t$ **then**
    $w_{t+1} \leftarrow w_t + y_t x_t$
  else
    $w_{t+1} \leftarrow w_t$
  end if
end for
return w_N



https://commons.wikimedia.org/wiki/File:Perceptron_example.svg

Perceptron: Mistake Bound

Theorem (Perceptron Mistake Bound)

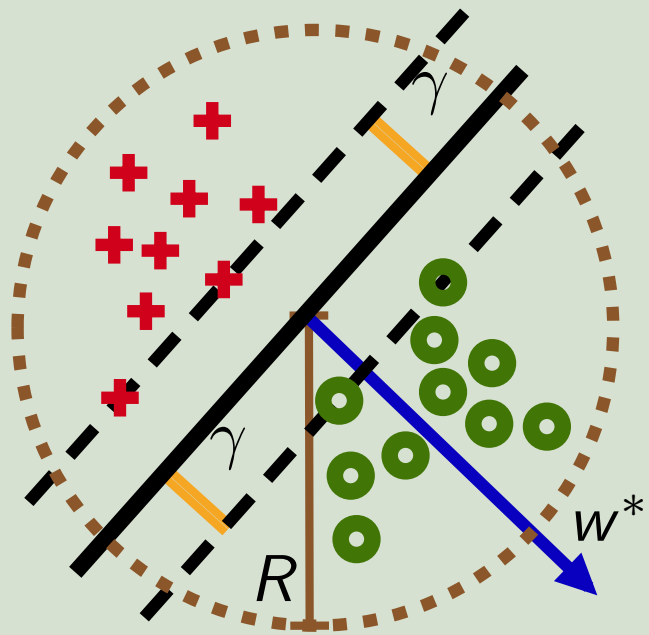
We need two assumptions for perceptron to work:

- *the data is* **linearly separable** *with margin* γ.
- *the input features have* **bounded norm**: $\|x\|_2 \leq R \; \forall x$

Then perceptron makes at most R^2/γ^2 *mistakes during training:*

$$\sum_{t=0}^{N-1} \mathbb{1}\left(sign\langle w_t, x_t\rangle\right) \neq y_t) \leq \frac{R^2}{\gamma^2}.$$

See bonus slides for proof.



Perceptron: First Risk Bound

Consider doing **one pass** through the data to get $\{w_0, \ldots, w_{n-1}\}$.

The **expected** risk if we use $w' \sim \text{Uniform}(\{w_0, \ldots, w_{n-1}\})$ is

$$R(\hat{f}_{w'}) = \mathbb{E}_{p(x,y)}\mathbb{E}_{\mathcal{D}}\mathbb{E}_t\left[\mathbb{1}(\text{sign}(\langle w_t, x\rangle) \neq y)\right].$$

Renaming (x, y) to be (x_t, y_t),

$$\begin{aligned} R(\hat{f}_{w'}) &= \mathbb{E}_{\mathcal{D}}\mathbb{E}_t\left[\mathbb{1}(\text{sign}(\langle w_t, x_t\rangle) \neq y_t)\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\frac{1}{n}\sum_{t=0}^{n-1}\mathbb{1}(\text{sign}(\langle w_t, x_t\rangle) \neq y_t)\right] \end{aligned}$$

This is the number of mistakes perceptron makes during training!

$$\begin{aligned} &\leq \mathbb{E}_{\mathcal{D}}\left[\frac{1}{n}\frac{R^2}{\gamma^2}\right] && \text{(by the mistake bound)} \\ &= \frac{1}{n}\frac{R^2}{\gamma^2} \end{aligned}$$

Perceptron: Second Risk Bound

Now, let's use the **final** w_N obtained by iterating through $\mathcal{D}$ until all examples are correctly classified.

$$R(\hat{f}_{w_N}) = \mathbb{E}_{p(x,y)}\mathbb{E}_{\mathcal{D}}\left[\mathbb{1}(\text{sign}(\langle w_N, x\rangle) \neq y)\right].$$

Again, rename (x, y) to be a new example (x_n, y_n) for $\mathcal{D}$.

$$R(\hat{f}_{w_N}) = \mathbb{E}_{\mathcal{D}\cup(x_n,y_n)}\left[\mathbb{1}(\text{sign}(\langle w_N, x_n\rangle) \neq y_n)\right].$$

Let w_N^{-j} be the weights obtained without example (x_j, y_j):

$$R(\hat{f}_{w_N}) = \mathbb{E}_{\mathcal{D}\cup(x_n,y_n)}\left[\frac{1}{n+1}\sum_{j=0}^{n}\mathbb{1}(\text{sign}(\langle w_N^{-j}, x_j\rangle) \neq y_j)\right].$$

$$\leq \left(\frac{1}{n+1}\right)\frac{R^2}{\gamma^2}. \qquad \text{(by the mistake bound)}$$

Perceptron: Conclusions

Random Selection		**Final Weights**
$\left(\frac{1}{n}\right) \frac{R^2}{\gamma^2}$	v.s.	$\left(\frac{1}{n+1}\right) \frac{R^2}{\gamma^2}$

Perceptron shows us:

- Risk has a complex dependence on the **parameterization**:
  - R/γ somehow measures the model capacity.
- Risk has a complex dependence on **optimization**:
  - Exactly optimizing perceptron gives only minor improvement.

Why Does Deep Learning Work?

Deep Learning: Different Stories

There are two deep learning stories.

What We Expect	**What We See**
More Parameters	More Parameters
$\Downarrow$	$\Downarrow$
Higher Model Capacity	??
$\Downarrow$	$\Downarrow$
More Overfitting	Better Generalization

Take-Home Message: model capacity is not just parameters!

Deep Learning: Filling in the Gap

We're quickly filling in the gap with possible sources of implicit and explicit regularization.

What's Actually Happening

More Parameters

$\Downarrow$

$\{$SGD, Architecture, Dropout, L2, Sharp Local Minima, Interpolation$\}$

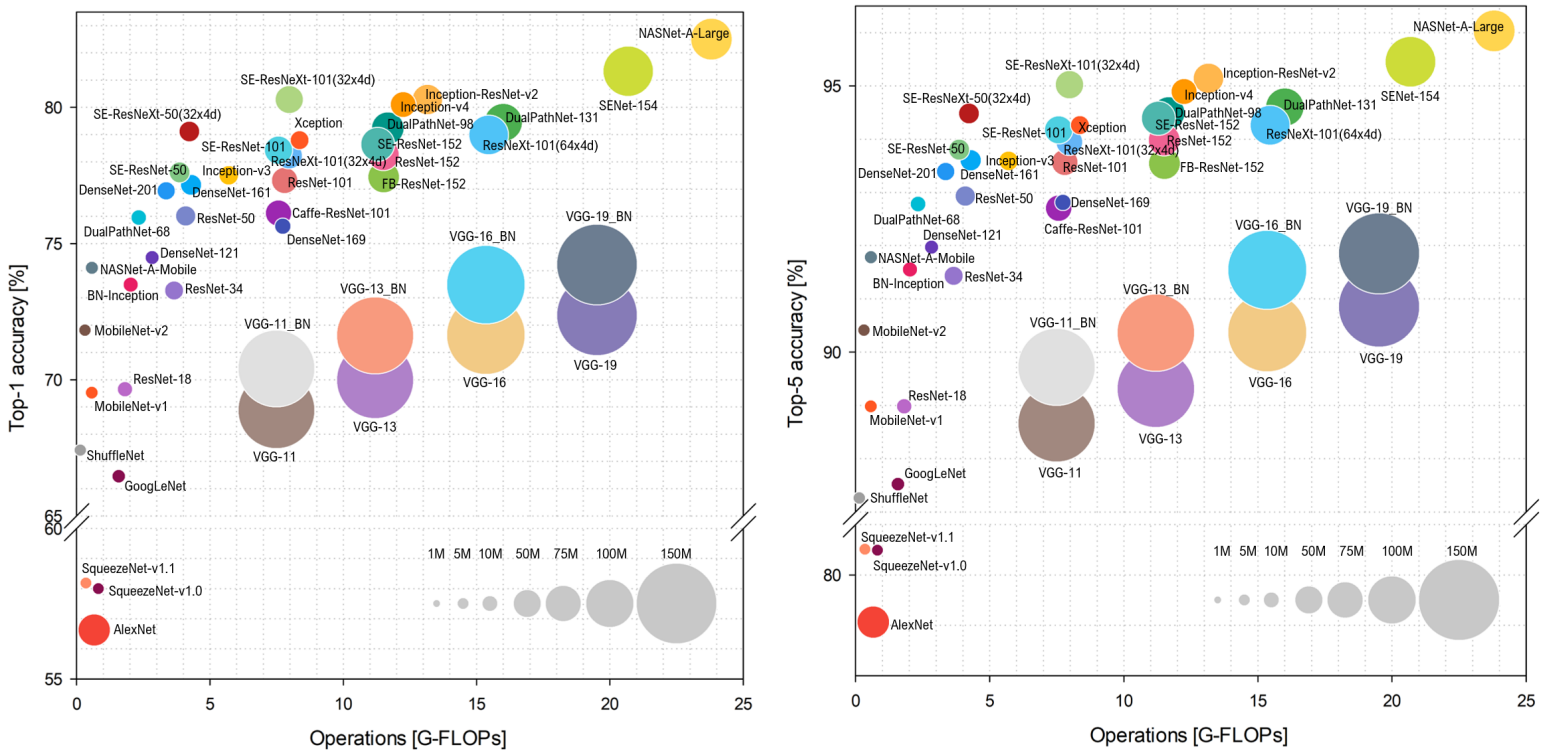
$\Downarrow$

Controlled Capacity

$\Downarrow$

Better Generalization

Deep Learning: Frontiers of Research



- **Sharp vs Flat Minima**: Some local minima generalize much better than others.
- **Implict Bias of SGD**: SGD regularizes towards particular solutions that generalize well.
- **Interpolation**: Highly over-parameterized models don't obey traditional bias-variance tradeoff.

Summary

Here's what we discussed today:

- Deep neural networks work very well for a variety of problems.
- Making neural networks bigger improves performance even when training accuracy has saturated.
- Number of parameters may be a poor measure of capacity.
- New research looks at the capacity of neural networks via
  - types of local minima,
  - properties of optimization procedures,
  - the role of over-parameterization/interpolation.

Signup Sheet

Acknowledgements

- The perceptron example and analysis comes from Sasha Rakhlin and Peter Bartlett.
  - See their excellent series on generalization from the Simons Institute.

Bonus Slides

Bonus: Perceptron Mistake Bound (1)

Theorem (Perceptron Mistake Bound)

We need two assumptions for perceptron to work:

- *the data is* **linearly separable** *with margin* γ.
- *the input features have* **bounded norm**: $\|x\|_2 \leq R \; \forall x$

Then perceptron makes at most R^2/γ^2 *mistakes during training:*

$$\sum_{t=0}^{N-1} \mathbb{1}\left(sign\langle w_t, x_t\rangle\right) \neq y_t) \leq \frac{R^2}{\gamma^2}.$$

Starting Place: The proof focuses on the angle between the normal vector for a max-margin hyperplane w^* and w_t:

$$\langle w_t, w_* \rangle \leq \|w_t\|_2 \|w^*\|_2$$

Bonus: Perceptron Mistake Bound (2)

Let $\{0 \dots T-1\}$ be iterations where perceptron makes a mistake.

Proof.

1. Margin of $\gamma \Rightarrow \exists w^* \in \mathbb{R}^d,\ y_i * \langle w^*, x_i \rangle \geq \gamma \|w^*\|_2 = \gamma$
   - Since the margin between any x and the hyperplane given by w^* is $\langle w^*, x \rangle / \|w^*\|_2$.
2. WLOG, let $\|w^*\|_2 = 1$ (rescaling doesn't affect hyperplanes).
3. $\langle w_T, w^* \rangle = \langle 0, w^* \rangle + \sum_{t=0}^{T-1} y_t \langle x_t, w^* \rangle \geq T * \gamma$
4. $\|w_T\|^2 = \|w_{T-1}\|^2 + \langle w_{T-1}, y_{T-1} x_{T-1} \rangle + \|x_{T-1}\|^2 \leq \|w_{T-1}\| + R^2$
5. By recursion on (4), $\|w_T\|^2 \leq T * R^2$
6. $\langle w_T, w^* \rangle \leq \|w_T\| \|w^*\| \Rightarrow T * \gamma \leq T^{1/2} * R \Rightarrow T \leq R^2 / \gamma^2$

□

Bonus: Towards Generalization Bounds

Intuitive Definition: $\hat{f}$ generalizes well if the empirical risk is a good approximation of the risk:

$$R_{\mathcal{D}}(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\hat{f}(x_i), y_i) \approx \mathbb{E}_{p(x,y)}\left[\mathcal{L}(\hat{f}(x), y)\right] = R(\hat{f}).$$

Formal(ish) Definition: $\hat{f}$ generalizes well if there are $\epsilon, \delta > 0$ such that

$$Pr_{\mathcal{D}}\left(R_{\mathcal{D}}(\hat{f}) \geq R(\hat{f}) + \epsilon\right) \leq \delta.$$

- $\hat{f}$ is a random variable because $\mathcal{D}$ is random.
- $\hat{f}$ is good with high-probability if the empirical risk is small.
  - roughly the idea of "probably, approximately correct" learning.

References I


Ross Girshick.
Fast r-cnn object detection with caffe.
Microsoft Research, 2015.


Tero Karras, Samuli Laine, and Timo Aila.
A style-based generator architecture for generative adversarial networks.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.


Jonathan Long, Evan Shelhamer, and Trevor Darrell.
Fully convolutional networks for semantic segmentation.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.


Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar.
Do imagenet classifiers generalize to imagenet?
arXiv preprint arXiv:1902.10811, 2019.