Understanding Deep Learning Requires Rethinking Generalization

Chiyuan Zhang
Massachusetts Institute of Technology
chiyuan@mit.edu

Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht
University of California, Berkeley
brecht@berkeley.edu

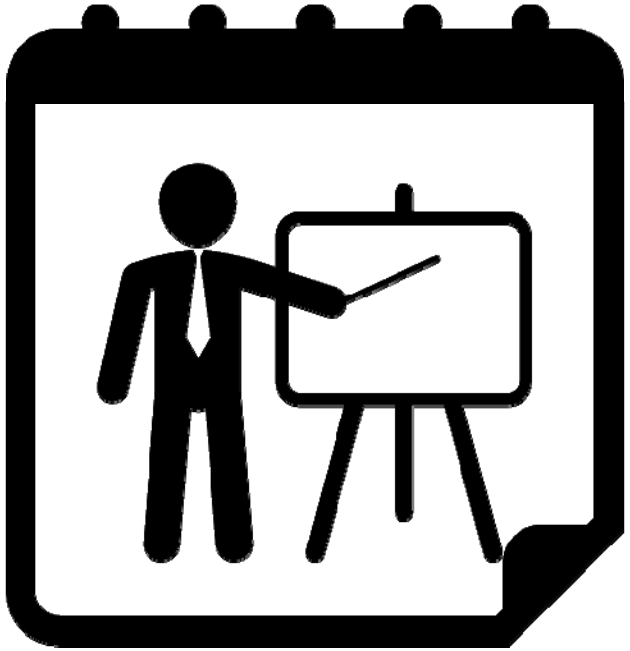
Oriol Vinyals
Google DeepMind
vinyals@google.com

ICLR Best Paper Award 2017

Presentation by **Rodney LaLonde** at the University of Central Florida's (UCF) Center for Research in Computer Vision (CRCV).

Presentation Outline

- Motivation
- Background
- Experimental Findings
- Discussion





Question:

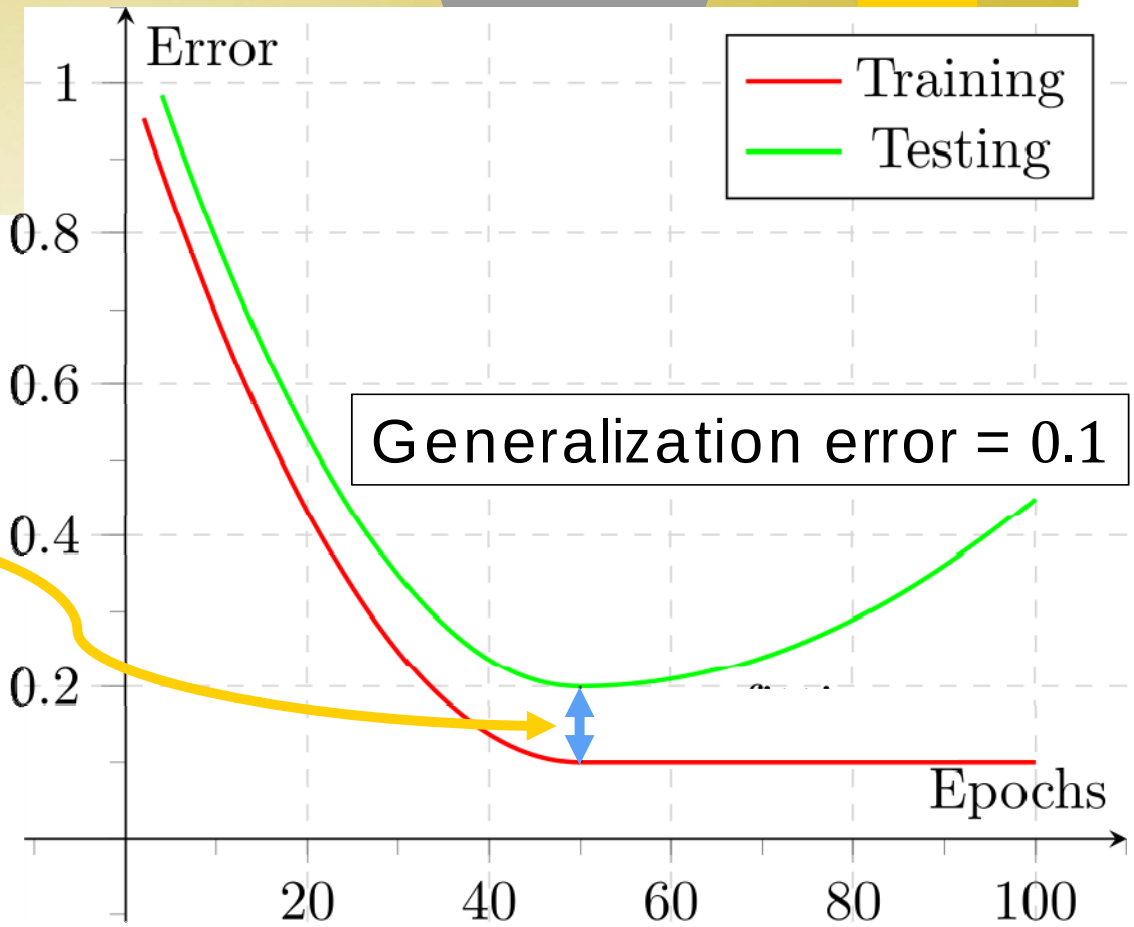
"What is it then that distinguishes neural networks that **generalize well** from those that **don't**?"

Background

- Generalization Error
- Model Capacity
- Regularization

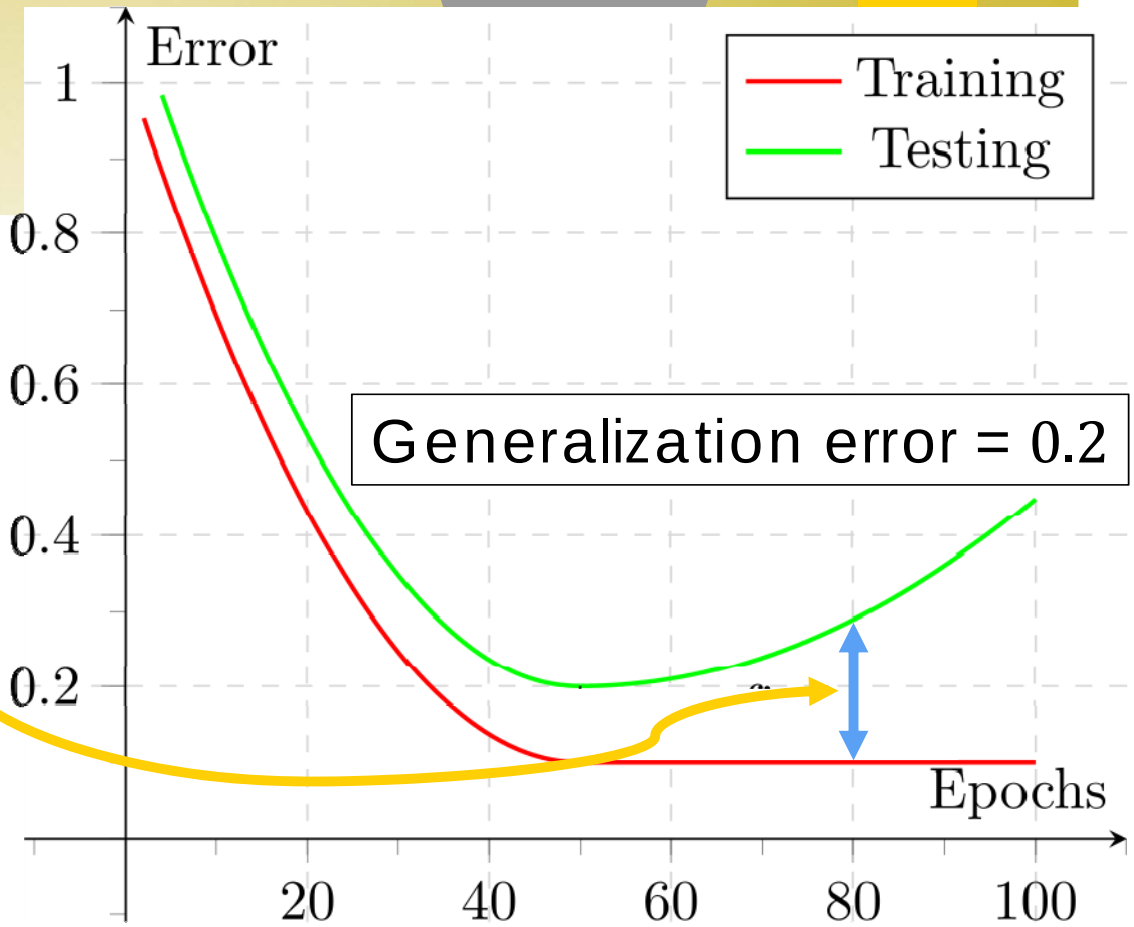
Generalization Error

The difference between **training error** and **test error**



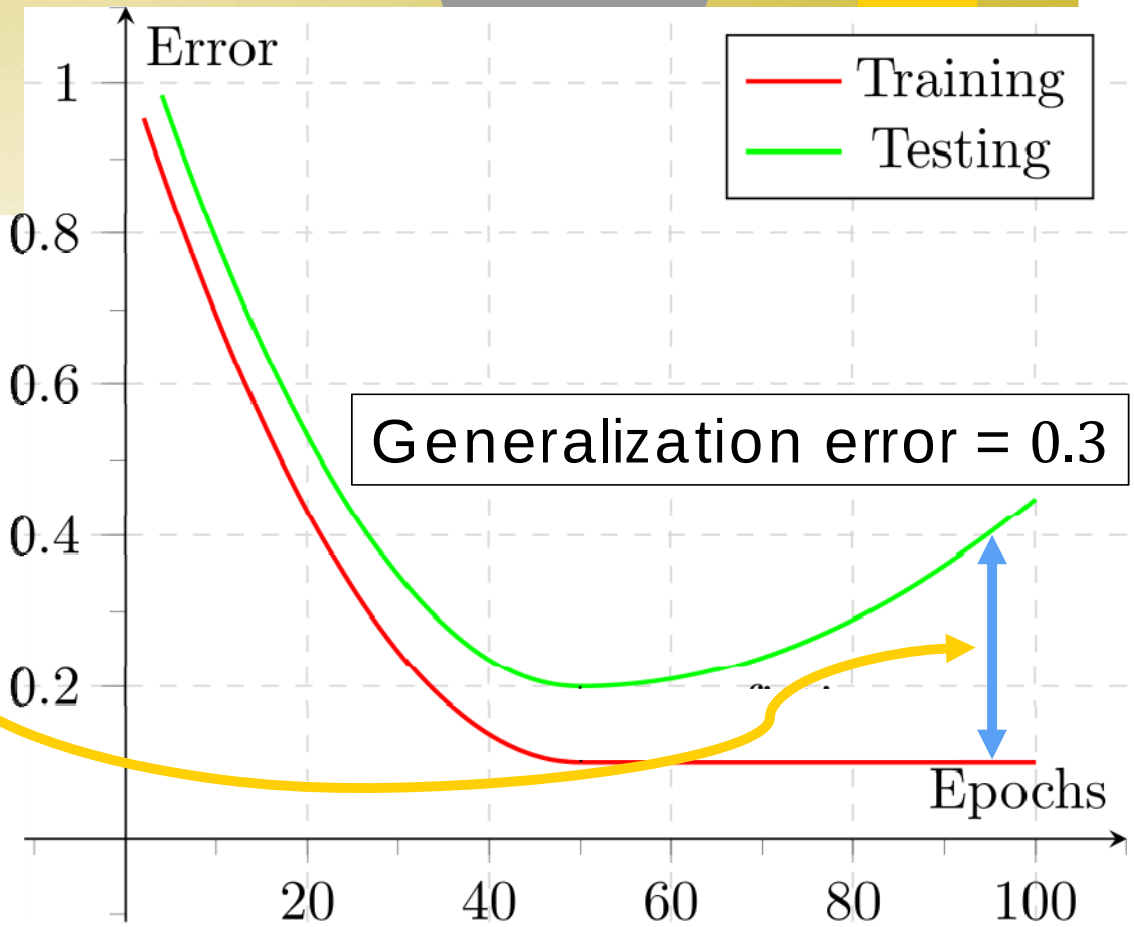
Generalization Error

The difference between **training error** and **test error**



Generalization Error

The difference between **training error** and **test error**



Universal Approximation Theorem

- A feed-forward network with a **single hidden layer** containing a finite number of neurons, can **approximate any continuous function** on compact subsets of $\mathbb{R}^n$.*
  - George Cybenko proved this for sigmoid activation functions.[1]
- Does not define the algorithmic learnability of those parameters.

* Some mild assumptions about the activation function must be met.

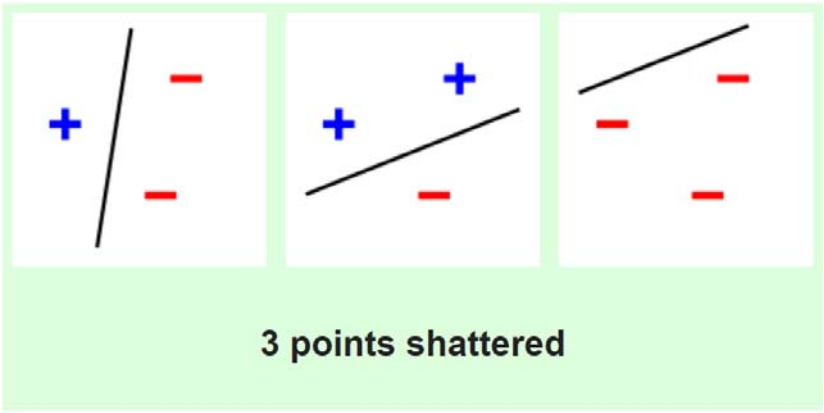
[1] Cybenko, G. (1989) "Approximations by superpositions of sigmoidal functions", Mathematics of Control, Signals, and Systems, 2 (4), 303-314.

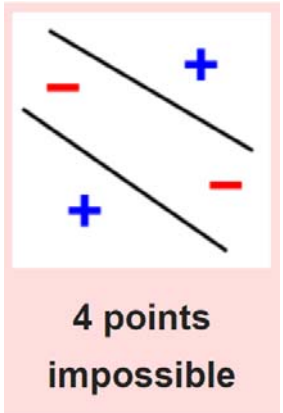
Vapnik-Chervonenkis theory
VC-Dimension

- A classification model $\boldsymbol{f}$ with some parameter vector $\boldsymbol{\theta}$ is said to ***shatter*** a set of data points $(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n})$ if, for all assignments of labels to those points, there exists a $\boldsymbol{\theta}$ such that the model $\boldsymbol{f}$ makes **no errors** when evaluating that set of data points.

Vapnik-Chervonenkis theory
VC-Dimension

- The **VC dimension** of a model f is the **maximum number of data points** that can be arranged so that f shatters them.





VC-Dimension – Statistical Learning Theory

- Probabilistic upper bound on test error:
- $$\mathrm{P}\left(error_{test} \leq error_{training} + \sqrt{\frac{1}{N}\left[D\left(log\left(\frac{2N}{D}\right)+1\right) - log\left(\frac{\eta}{4}\right)\right]}\right) = 1-\eta$$
- Valid only when $D \ll N$.
- Not useful for deep neural networks where typically $D \gg N$.

Regularization

- Explicit Regularization
  - Weight Decay
  - Dropout
  - Data Augmentation
- Implicit Regularization
  - Early Stopping
  - Batch Normalization
  - SGD

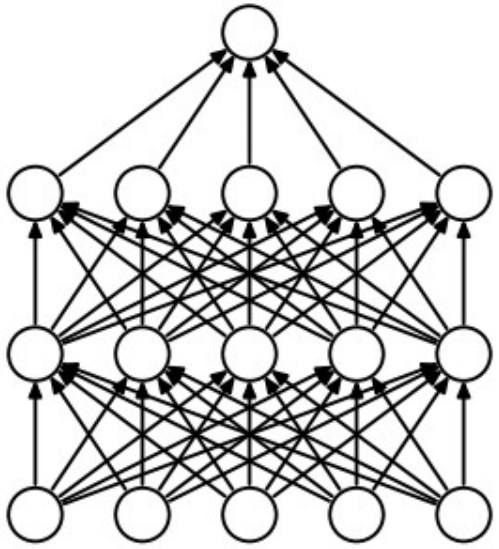
L2 Regularization – "Weight Decay"

- Standard weight update: $w_{t+1} \leftarrow w_t - \eta_t \frac{\partial L(\boldsymbol{w})}{\partial w_t}$
- $L(w) = L_0(w) + \frac{\lambda}{2} w^2$ *
- New weight update: $w_{t+1} \leftarrow w_t - \eta_t \frac{\partial L_0(\boldsymbol{w})}{\partial w_t} - \eta_t \lambda w_t$ *
- Forces the weights to become small, "decay".

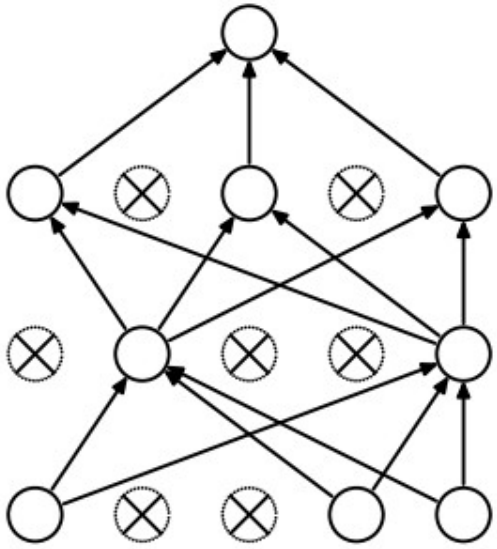
* Krogh, Anders, and John A. Hertz. "A simple weight decay can improve generalization." In Advances in neural information processing systems, pp. 950-957. 1992.

Dropout

- Randomly drop neurons from layers in the network.
- Removes reliance on individual neurons.



(a) Standard Neural Net

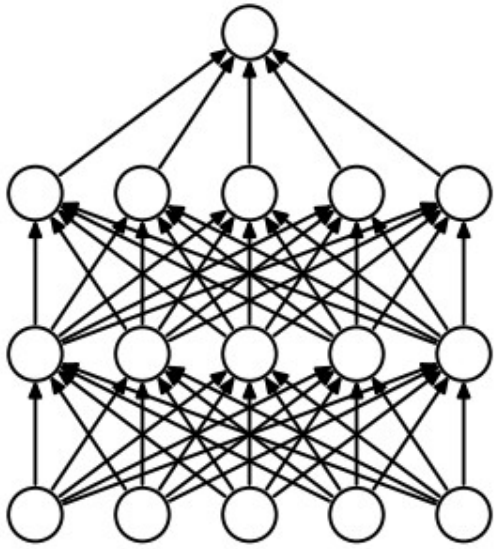


(b) After applying dropout.

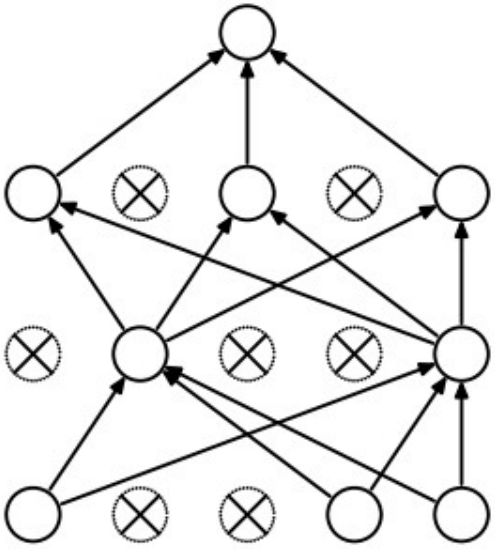
Figure taken from: Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15, no. 1 (2014): 1929-1958.

Dropout

- Learns redundancies.
- Learns more nuanced set of feature detectors.



(a) Standard Neural Net



(b) After applying dropout.

Figure taken from: Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15, no. 1 (2014): 1929-1958.

Data Augmentation

- Domain specific transformations of the input data.
- For images: Shown in figure (also random noise, sheer, zoom, elastic deformations, etc.)

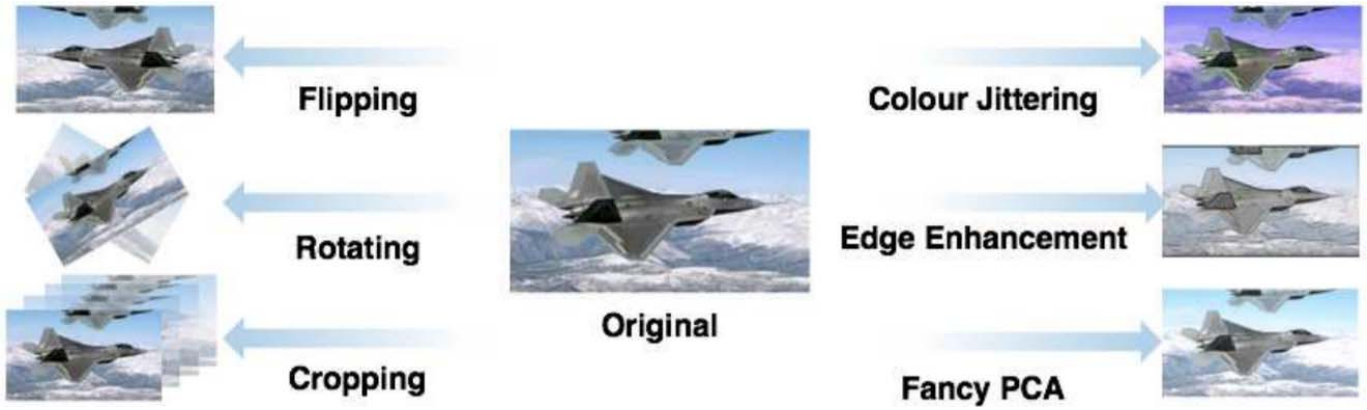


Figure from: Taylor, Luke, and Geoff Nitschke. "Improving Deep Learning using Generic Data Augmentation." arXiv preprint arXiv:1708.06020 (2017).

Data Augmentation

- Increases the input space (*i.e.* all possible images we care about).

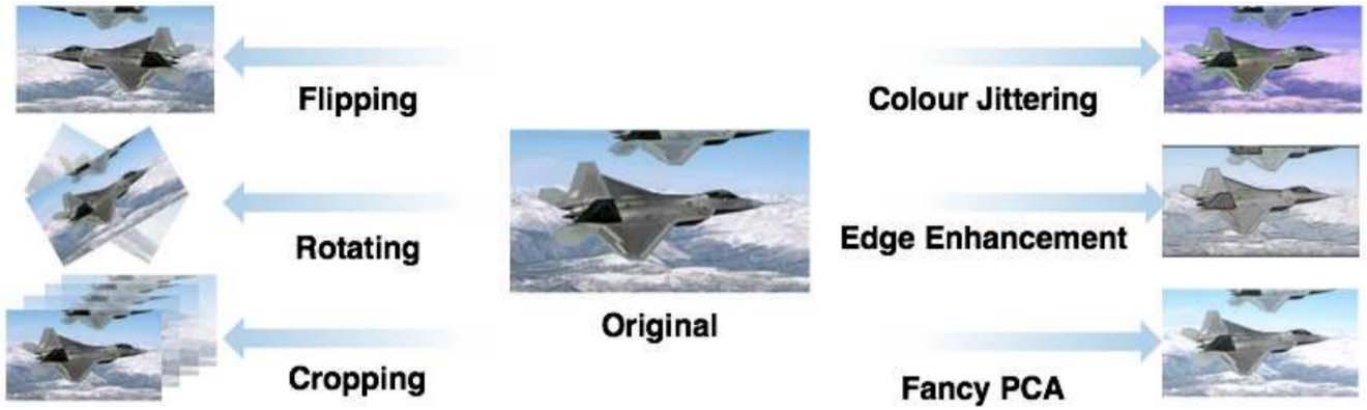
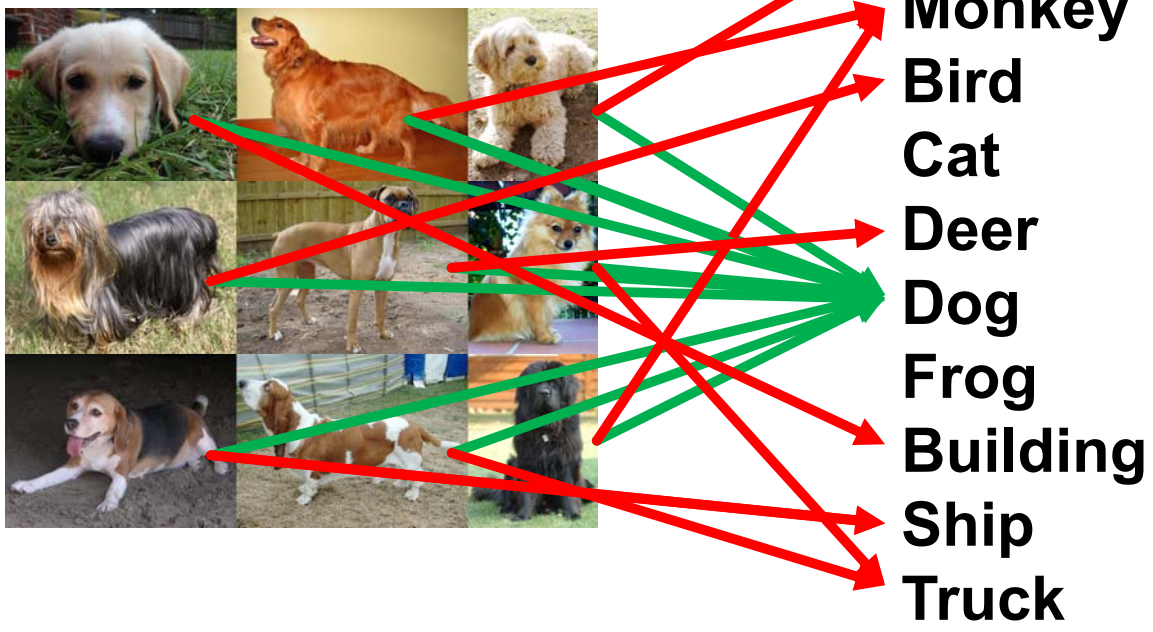


Figure from: Taylor, Luke, and Geoff Nitschke. "Improving Deep Learning using Generic Data Augmentation." arXiv preprint arXiv:1708.06020 (2017).

Experimental Findings

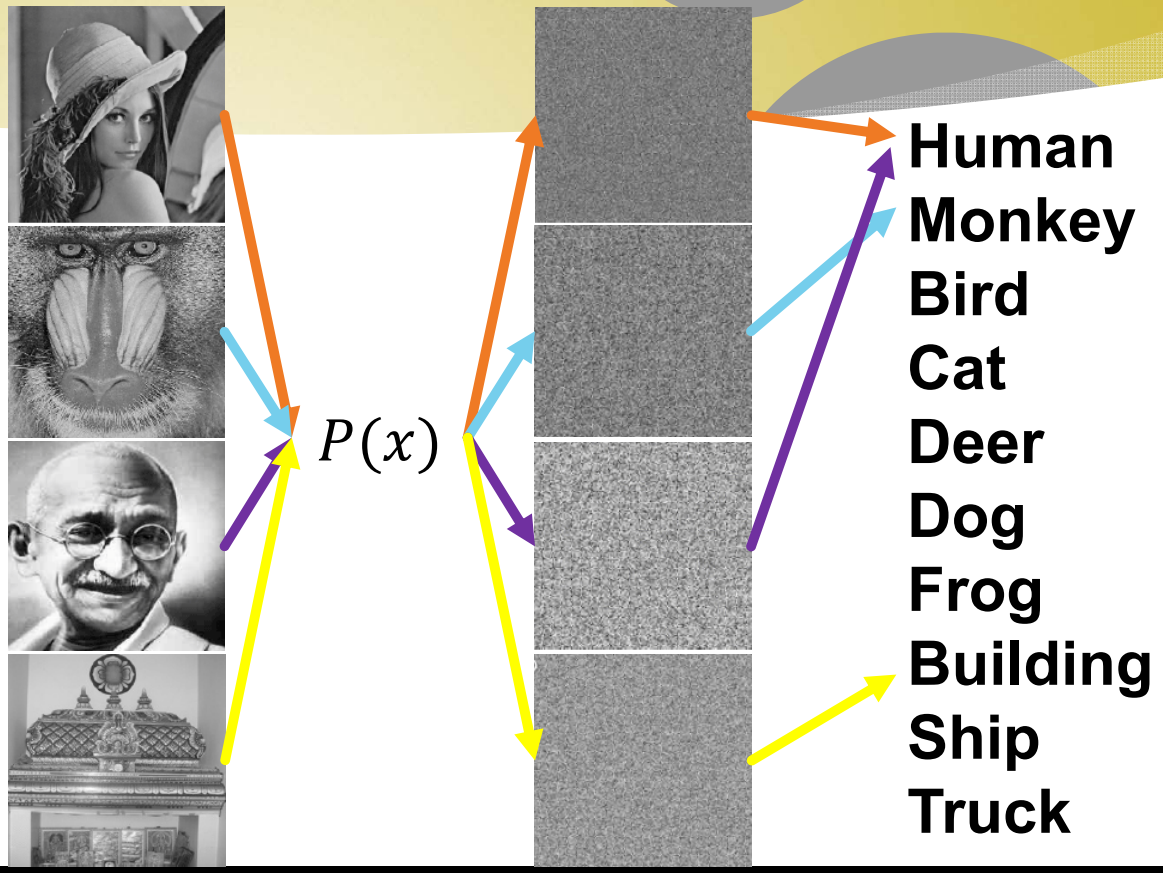
Randomization Tests

- True labels
- Partially corrupted labels
- Random labels



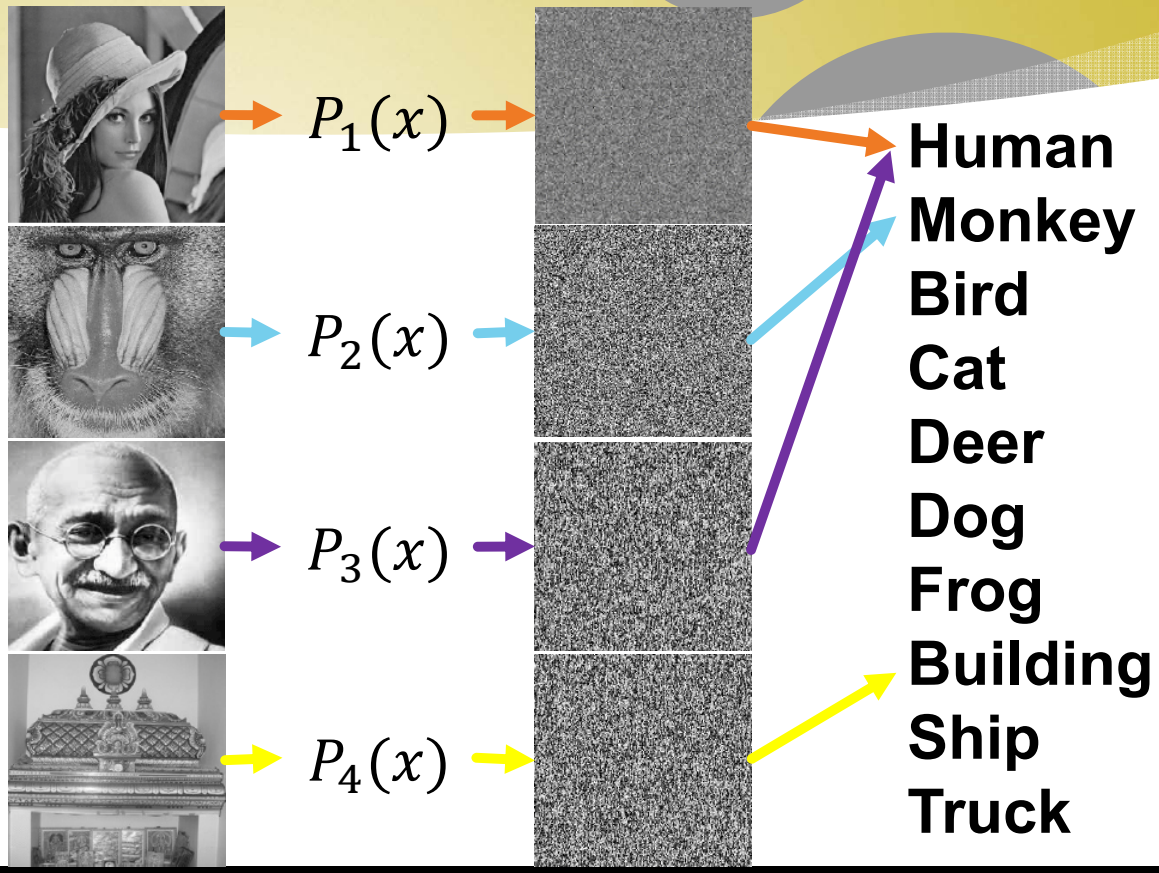
Randomization Tests

- True labels
- Partially corrupted labels
- Random labels
- Shuffled pixels



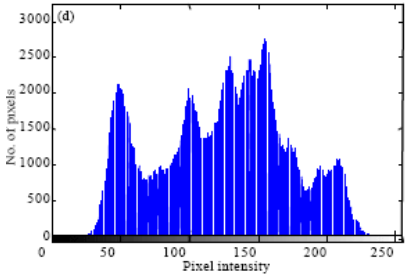
Randomization Tests

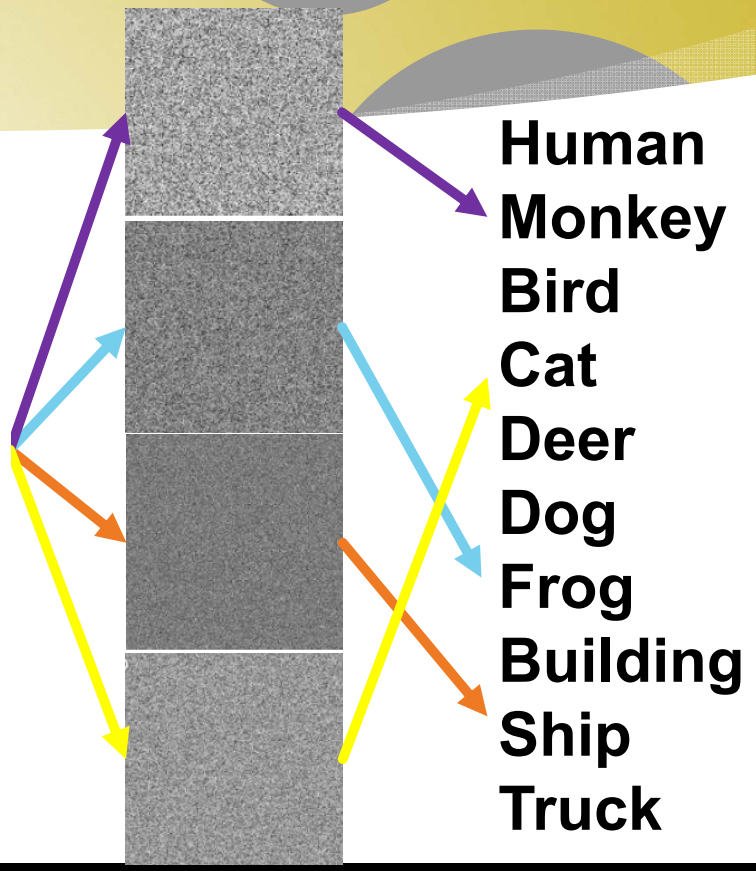
- True labels
- Partially corrupted labels
- Random labels
- Shuffled pixels
- Random pixels



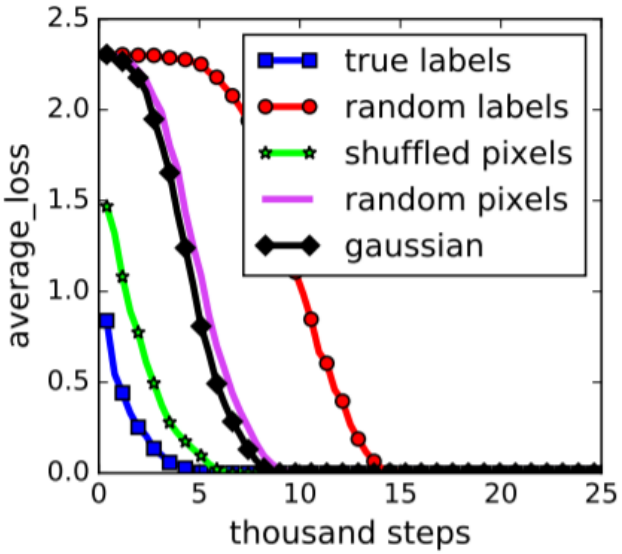
Randomization Tests

- True labels
- Partially corrupted labels
- Random labels
- Shuffled pixels
- Random pixels
- Gaussian

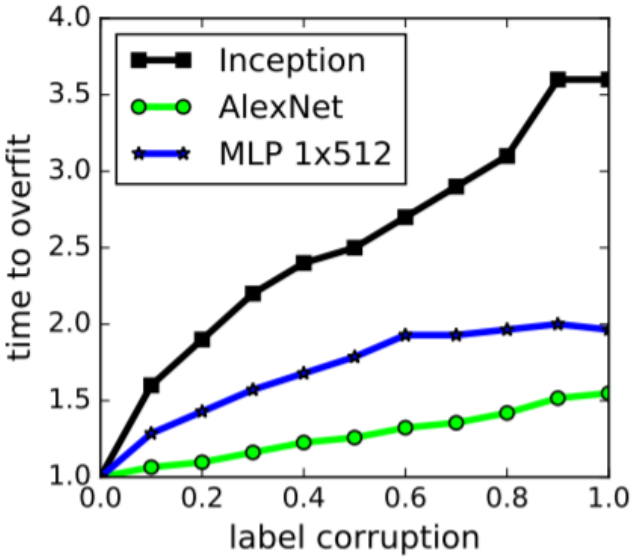




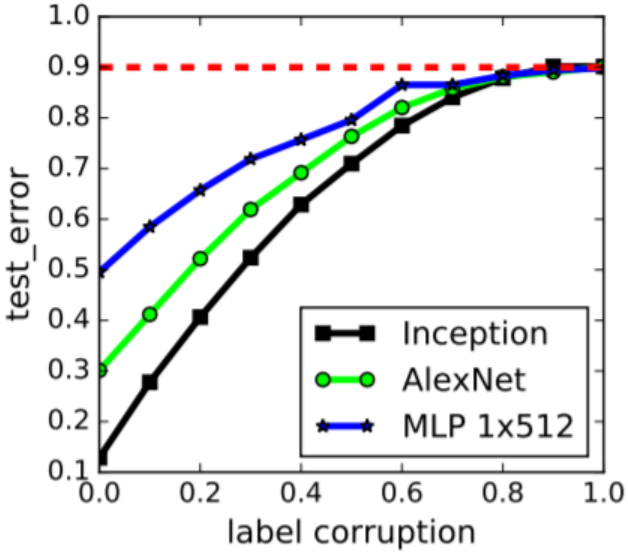
Results of Randomization Tests



(a) learning curves



(b) convergence slowdown

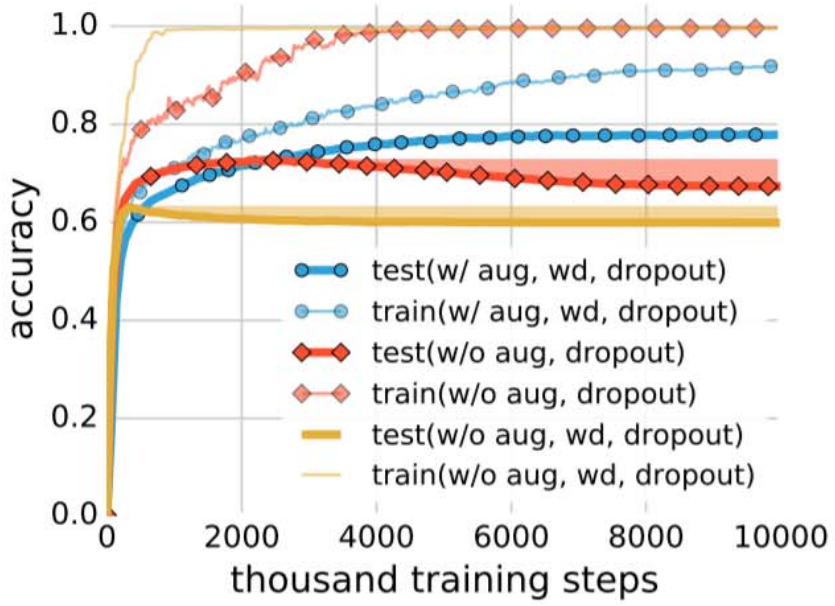


(c) generalization error growth

Conclusions & Implications

- Conclusion:
  - Deep neural networks **easily fit random labels**.
- Implications:
  - The **effective capacity** of neural networks is **sufficient for memorizing** the entire data set.
  - Even **optimization** on random labels **remains easy**.

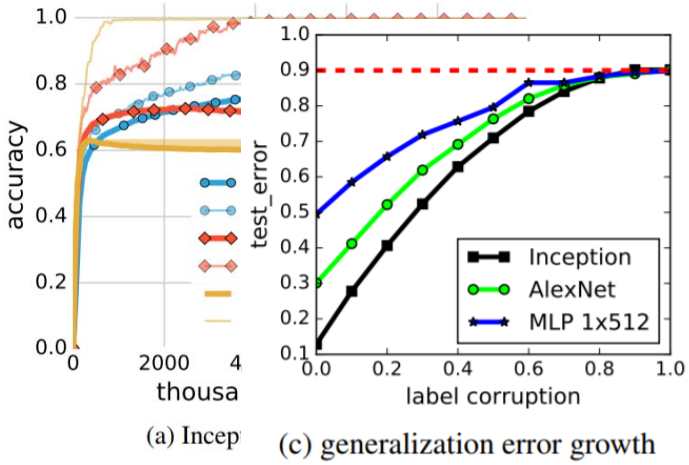
Explicit Regularization Tests



(a) Inception on ImageNet

Conclusions & Implications

- Conclusions:
  - Augmenting data is more powerful than only weight decay.
  - Bigger gains by **changing the model arch**
- Implications:
  - Explicit regularization may improve gene**neither necessary nor by itself sufficient**.

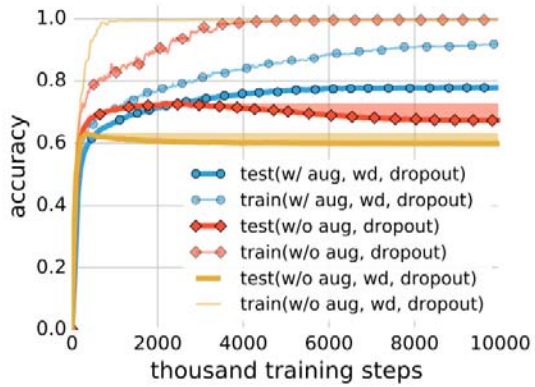


(a) Incep

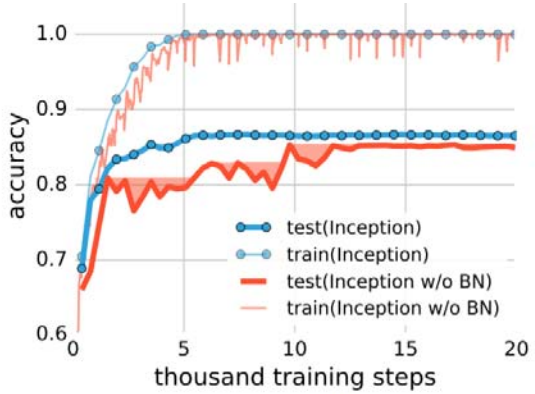
(c) generalization error growth

Implicit Regularization Findings

- Early stopping could potentially improve generalization.
- Batch normalization improves generalization.



(a) Inception on ImageNet



(b) Inception on CIFAR10

> “Both explicit and implicit regularizers… could help to improve the generalization performance. However, it is **unlikely that the regularizers are the fundamental reason for generalization**…”

Authors’ Conclusions

Finite-Sample Expressivity of Neural Networks

- At the "population level", depth k networks typically more powerful than depth $k-1$ networks.
- Given a finite sample size n, even a **two-layer neural network** can represent **any function** once the number of parameters p exceeds n.

Finite-Sample Expressivity of Neural Networks

- **Theorem 1**:
  - "There exists a two-layer neural network with ReLU activations and $2n + d$ weights that can represent any function on a sample of size n in d dimensions."

Finite-Sample Expressivity of Neural Networks

- A network C can represent any function of a sample size n in d dimensions if:
  - for every sample $S \subseteq \mathbb{R}^d$ with $|S| = n$ and
  - every function $f: S \to \mathbb{R}$,
- there exists a setting of weights of C such that $C(x) = f(x)$ for every $x \in S$.
- Can be extended to depth k networks with width $O\left(\frac{n}{k}\right)$.

Appeal to Linear Models

- Imagine n data points $\{(x_i, y_i)\}$, where x_i are d-dimensional feature vectors and y_i are labels.
- Solve: $\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} loss(w^T x_i, y_i)$. Eq. (2)
- If $n \geq d$ we can fit any labeling.

Appeal to Linear Models

- Let X denote the $n \times d$ matrix whose i-th row is x_i^T.
- If X has rank n, then $Xw = y$ has an **infinite number of solutions**.
- We find a **global min** of Eq. (2) by **solving this linear system**.

Investigating SGD

- Given SGD: $w_{t+1} = w_t - \eta_t e_t x_i$ and $w_0 = 0$,
  then $w = \sum_1^n \alpha_i x_i$ for some coefficients α.
- Therefore $w = X^T \alpha$ lies in the span of the data points.
- Replacing this and perfectly interpolating the labels,
  gives $Xw = y \Rightarrow XX^T \alpha = y$, which has a unique solution.

Investigating SGD

- Forming the kernel matrix (Gram matrix) $\mathrm{K} = XX^T$ and solving $\mathrm{K}\alpha = y$ for α yields a perfect fit on the labels.
- Turns out, this kernel solution is exactly the **minimum l_2-norm** solution to $Xw = y$.
- Hence **SGD converges to the solution with minimum norm**.
- Minimum norm is **not** predictive of generalization performance.

Final Conclusions

- **Effective capacity** of neural networks.
- Successful neural networks are large enough to **shatter the training data**.
- **Optimization** continues to be **easy** even when generalization is poor.

Final Conclusions

- SGD **may** be performing **implicit regularization** by converging to solutions with minimum l_2-norm.

- **Traditional measures** of model complexity **struggle to explain the generalization** of large neural networks.

Thank You!

- Questions and Discussions?