

---

---

# Performance Improvements

---

---

# Performance Improvements

- Learning Curves
  - Batch Norm
  - Dropout + Regularization
  - Data Augmentation
  - Tuning
  - Embeddings
  - Continuous Learning
  - Callbacks
-

---

# How to interpret learning Curve

- New model or keep training?
  - Are the errors converging?
    - Yes → more data
    - No → new model
-

---

# Codealong

[https://github.com/nrkfeller/YCBS\\_notes/blob/master/C7\\_Learning\\_Curves.ipynb](https://github.com/nrkfeller/YCBS_notes/blob/master/C7_Learning_Curves.ipynb)

---

---

# Batch Norm

Reduces the chances of overfitting

- Standardize output of layer
- Rescale by learned parameters

This gives us

- Higher learning rates → faster learning
  - Regularizes the model
  - Improves accuracy
-

---

# Explained...

**Input:** Values of  $x$  over a mini-batch:  $\mathcal{B} = \{x_{1\dots m}\}$ ;

Parameters to be learned:  $\gamma, \beta$

**Output:**  $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

---

1. Get mean
2. Get variance
3. Rescale using mean and variance
4. Scale and shift with gamma and beta, which is learned

---

# Significant perf improvement

Model	Resolution	Crops	Models	Top-1 error	Top-5 error
GoogLeNet ensemble	224	144	7	-	6.67%
Deep Image low-res	256	-	1	-	7.96%
Deep Image high-res	512	-	1	24.88	7.42%
Deep Image ensemble	variable	-	-	-	5.98%
BN-Inception single crop	224	1	1	25.2%	7.82%
BN-Inception multicrop	224	144	1	21.99%	5.82%
BN-Inception ensemble	224	144	6	20.1%	<b>4.9%*</b>

Figure 4: *Batch-Normalized Inception comparison with previous state of the art on the provided validation set comprising 50000 images. \*BN-Inception ensemble has reached 4.82% top-5 error on the 100000 images of the test set of the ImageNet as reported by the test server.*

---

---

# Codealong

[https://github.com/nrkfeller/YCBS\\_notes/blob/master/C7\\_Batch\\_Normalization.ipynb](https://github.com/nrkfeller/YCBS_notes/blob/master/C7_Batch_Normalization.ipynb)

---



---

---

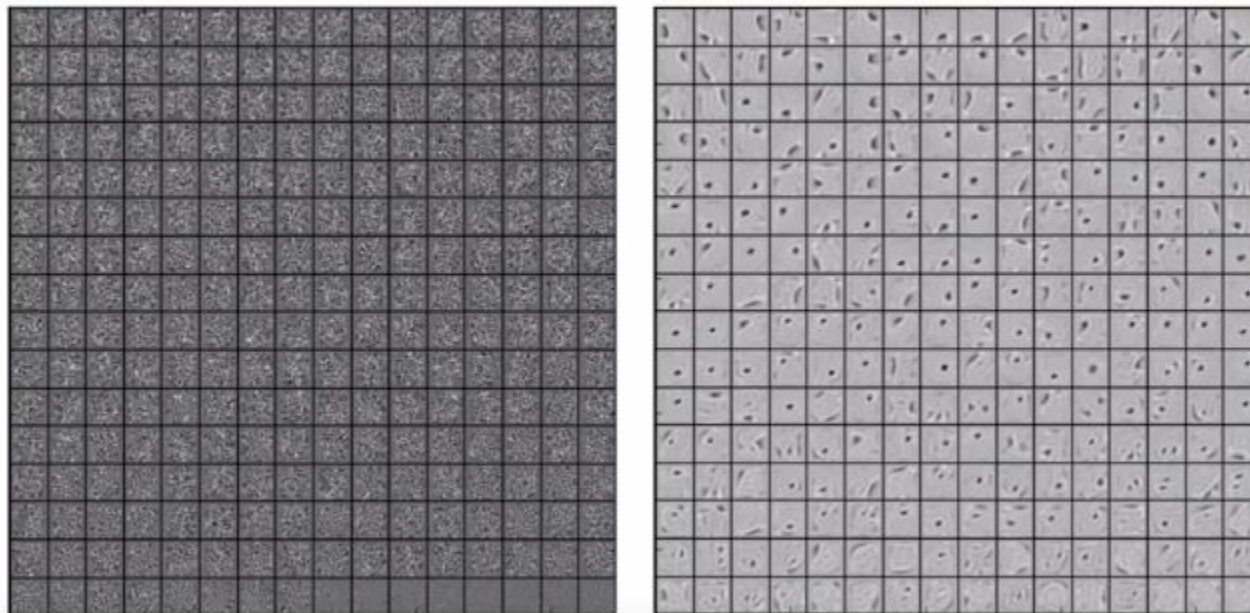
# Dropout

Quick recap (this should be understood)

- Build redundancy
  - Removes reliance on other nodes
  -
-

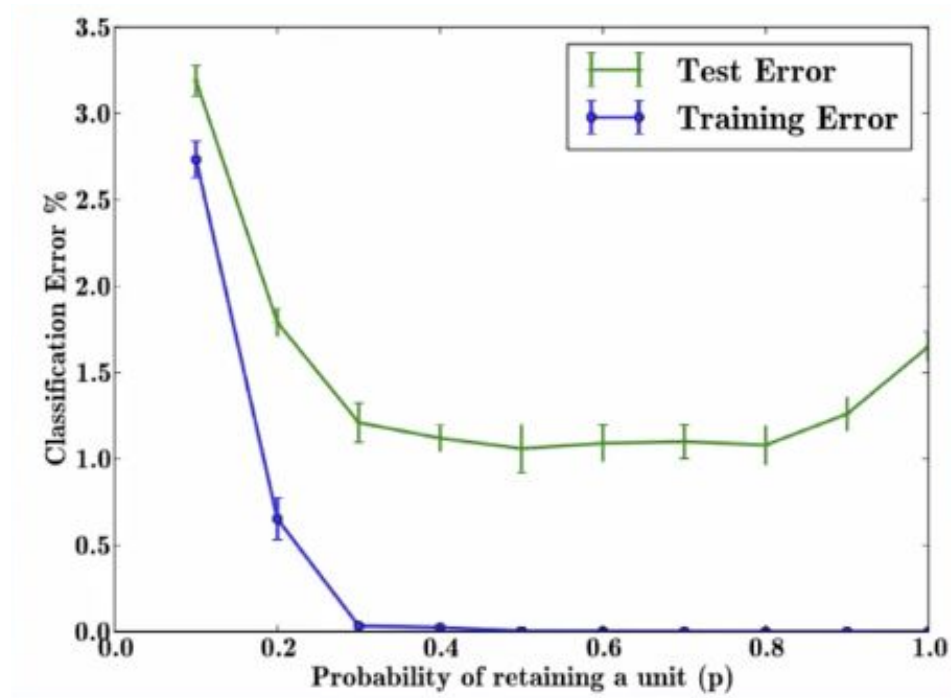
---

Noise robustness: right (dropout with  $p = 0.50$ )



---

**Only 1 Hyper parameter  $\rightarrow$  % dropout**  
**Interval of performance  $\sim .3$  to  $.6$  in this case**



---

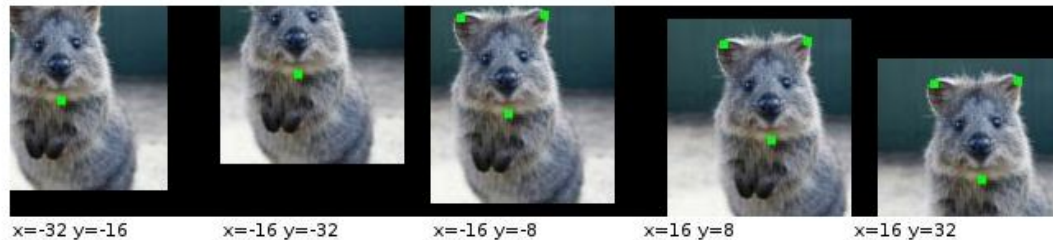
# Codealong

[https://github.com/nrkfeller/YCBS\\_notes/blob/master/C7\\_Dropout.ipynb](https://github.com/nrkfeller/YCBS_notes/blob/master/C7_Dropout.ipynb)

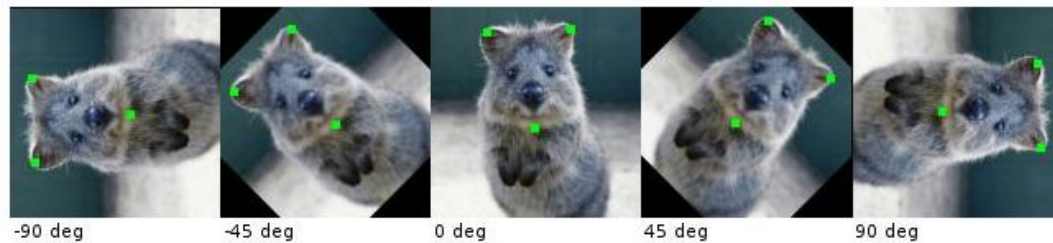
---

# Data Augmentation

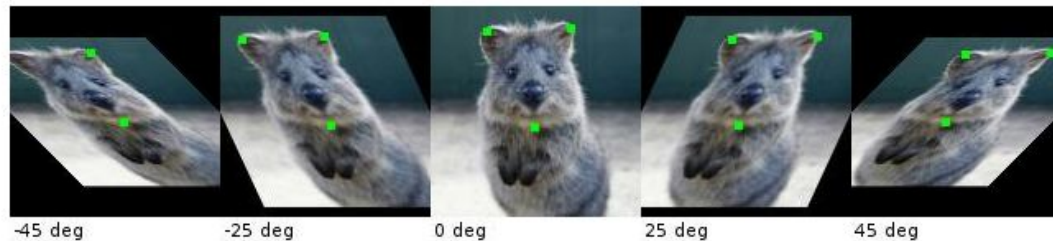
Affine: Translate



Affine: Rotate



Affine: Shear



---

---

# Codealong

[https://github.com/nrkfeller/YCBS\\_notes/blob/master/C7\\_Data\\_Augmentation.ipynb](https://github.com/nrkfeller/YCBS_notes/blob/master/C7_Data_Augmentation.ipynb)

---