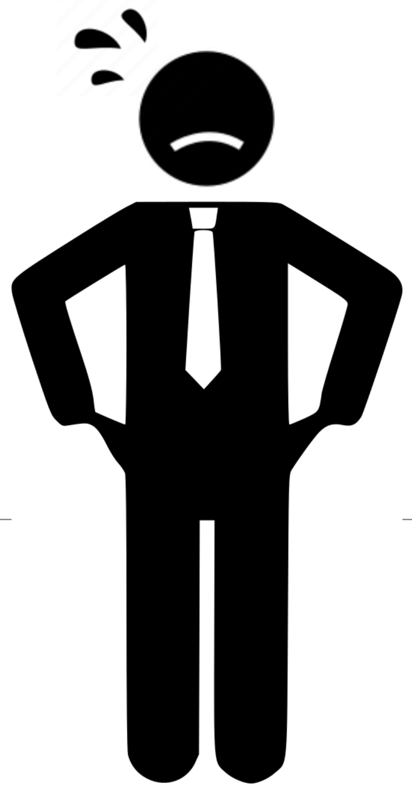
Deep Learning for Small Datasets

DAPHNÉ CHOPARD



Overall Goal (Supervised)

Given X and Y, and assuming there exists a true function $f(\cdot)$ such that

$$Y = f(X) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$$

the goal is to **estimate a model $\hat{f}(\cdot)$ that approximates $f(\cdot)$** such that

$$\left(Y - \hat{f}(X)\right)^2$$

is minimal.

Overall Goal

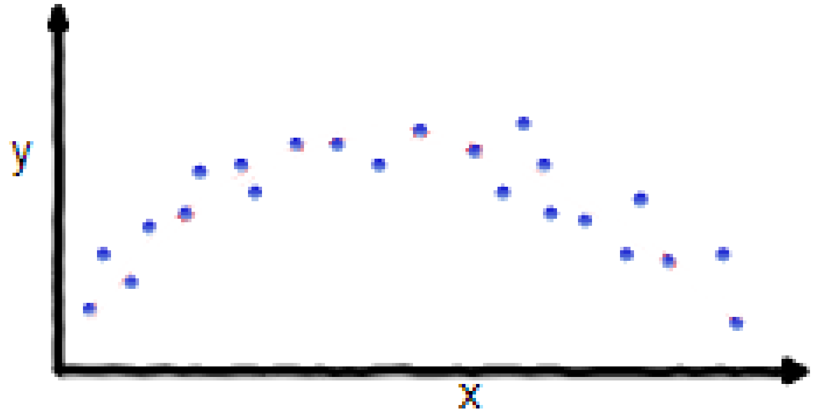
Estimate relationship between input X and output y:

$$\hat{f}(X) = y$$

Overall Goal

Estimate relationship between input X and output y:

$$\hat{f}(X) = y$$

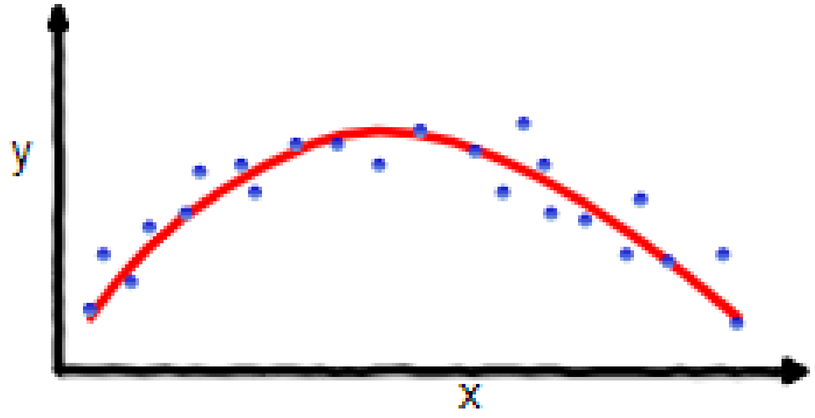


SOURCE: HTTPS://TOWARDSDATASCIENCE.COM/UNDERSTANDING-THE-BIAS-VARIANCE-TRADEOFF-165E6942B229

Overall Goal

Estimate relationship between input X and output y:

$$\hat{f}(X) = y$$



SOURCE: HTTPS://TOWARDSDATASCIENCE.COM/UNDERSTANDING-THE-BIAS-VARIANCE-TRADEOFF-165E6942B229

Expected Generalization Error

$$\mathrm{Err}(x) = E\left[\left(\left(Y - \hat{f}(x)\right)^2\right)\right]$$

Expected Error of Approximation

$$\mathrm{Err}(x) = E\left[\left(\left(Y - \hat{f}(x)\right)^2\right)\right]$$

$$= f(x)^2 + E[\hat{f}(x)^2] - 2E\left[\hat{f}(x)\right]f(x) + \sigma_\epsilon^2$$

Expected Error of Approximation

$$\mathrm{Err}(x) = E\left[\left(\left(Y - \hat{f}(x)\right)^2\right)\right]$$

$$= f(x)^2 + E[\hat{f}(x)^2] - 2E\left[\hat{f}(x)\right]f(x) + \sigma_\epsilon^2$$

$$= \quad f(x)^2 - 2E\left[\hat{f}(x)\right]f(x) + E\left[\hat{f}(x)^2\right] \quad + \sigma_\epsilon^2$$

Expected Error of Approximation

$$\mathrm{Err}(x) = E\left[\left(\left(Y - \hat{f}(x)\right)^2\right)\right]$$

$$= f(x)^2 + E[\hat{f}(x)^2] - 2E\left[\hat{f}(x)\right]f(x) + \sigma_\epsilon^2$$

$$= \boldsymbol{E}\left[\hat{\boldsymbol{f}}(\boldsymbol{x})\right]^{\mathbf{2}} + f(x)^2 - 2E\left[\hat{f}(x)\right]f(x) + E\left[\hat{f}(x)^2\right] - \boldsymbol{E}\left[\hat{\boldsymbol{f}}(\boldsymbol{x})\right]^{\mathbf{2}} + \sigma_\epsilon^2$$

Expected Error of Approximation

$$\mathrm{Err}(x) = E\left[\left(\left(Y - \hat{f}(x)\right)^2\right)\right]$$

$$= f(x)^2 + E[\hat{f}(x)^2] - 2E\left[\hat{f}(x)\right]f(x) + \sigma_\epsilon^2$$

$$= E\left[\hat{f}(x)\right]^2 + f(x)^2 - 2E\left[\hat{f}(x)\right]f(x) + E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2 + \sigma_\epsilon^2$$

$$= \left(E\left[\hat{f}(x)\right] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right] + \sigma_\epsilon^2$$

Expected Error of Approximation

$$\mathrm{Err}(x) = E\left[\left(\left(Y - \hat{f}(x)\right)^2\right)\right]$$

$$= f(x)^2 + E[\hat{f}(x)^2] - 2E[\hat{f}(x)]f(x) + \sigma_\epsilon^2$$

$$= E[\hat{f}(x)]^2 + f(x)^2 - 2E[\hat{f}(x)]f(x) + E[\hat{f}(x)^2] - E[\hat{f}(x)]^2 + \sigma_\epsilon^2$$

$$= \underbrace{\left(E[\hat{f}(x)] - f(x)\right)^2}_{\text{Squared Bias Error}} + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_\epsilon^2$$

Expected Error of Approximation

$$\mathrm{Err}(x) = E\left[\left(\left(Y - \hat{f}(x)\right)^2\right)\right]$$

$$= f(x)^2 + E[\hat{f}(x)^2] - 2E\left[\hat{f}(x)\right]f(x) + \sigma_\epsilon^2$$

$$= E\left[\hat{f}(x)\right]^2 + f(x)^2 - 2E\left[\hat{f}(x)\right]f(x) + E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2 + \sigma_\epsilon^2$$

$$= \underbrace{\left(E\left[\hat{f}(x)\right] - f(x)\right)^2}_{\text{Squared Bias Error}} + \underbrace{E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right]}_{\text{Variance Error}} + \sigma_\epsilon^2$$

Bias Error

- How far in general are the predictions from the correct value?

$$\text{Bias}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)\right] - f(x)$$

Bias Error

- How far in general are the predictions from the correct value?

$$\mathrm{Bias}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)\right] - f(x)$$

High bias ?

Bias Error

- How far in general are the predictions from the correct value?

$$\mathrm{Bias}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)\right] - f(x)$$

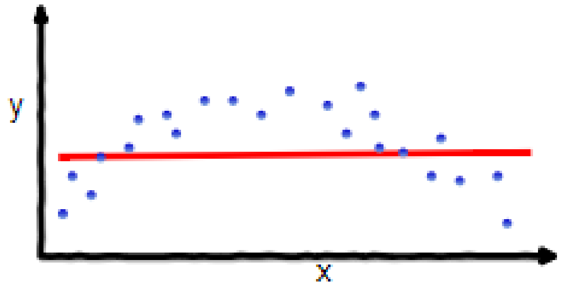
High bias ⇔ model too simple to fit well

Bias Error

- How far in general are the predictions from the correct value?

$$\text{Bias}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)\right] - f(x)$$

High bias ⬄ model too simple to fit well



Underfitting

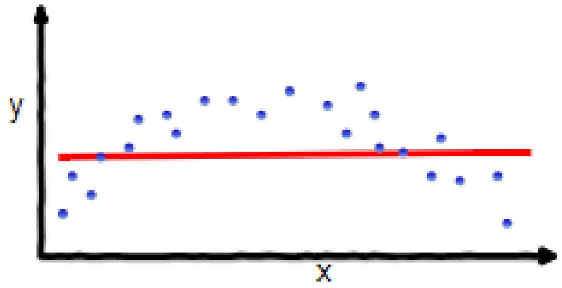
Bias Error

- How far in general are the predictions from the correct value?

$$\text{Bias}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)\right] - f(x)$$

High bias ⇔ model too simple to fit well

⇔ training error large



Underfitting

Bias Error

- How far in general are the predictions from the correct value?

$$\text{Bias}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)\right] - f(x)$$

High bias	⬄	model too simple to fit well
	⬄	training error large
Low bias	⬄	model complex enough to fit well
	⬄	training error low

Variance Error

- How much do the predictions vary between different model realizations?

$$\mathrm{Var}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2$$

Variance Error

- How much do the predictions vary between different model realizations?

$$\mathrm{Var}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2$$

High variance?

Variance Error

- How much do the predictions vary between different model realizations?

$$\mathrm{Var}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2$$

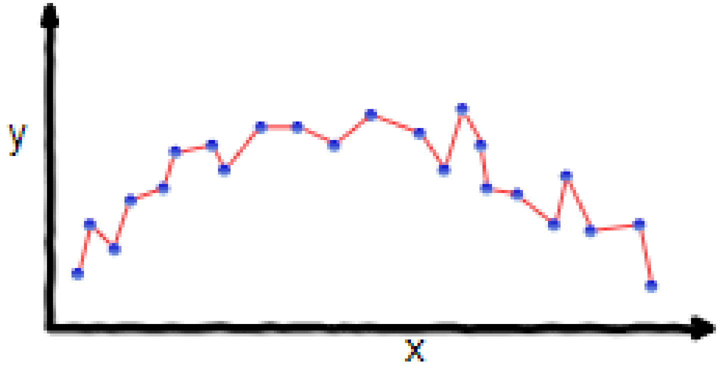
High variance ⇔ model too complex

Variance Error

- How much do the predictions vary between different model realizations?

$$\mathrm{Var}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2$$

High variance ⇔ model too complex



Overfitting

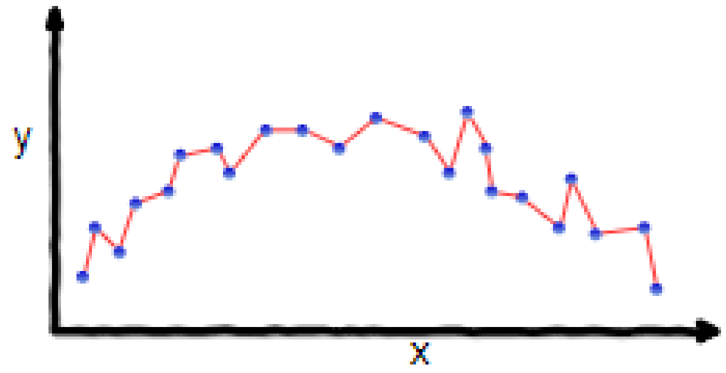
Variance Error

- How much do the predictions vary between different model realizations?

$$\mathrm{Var}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2$$

High variance ⬄ model too complex

⬄ validation error large



Overfitting

Variance Error

- How much do the predictions vary between different model realizations?

$$\mathrm{Var}\left(\hat{f}(x)\right) = E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2$$

High variance ⇔ model too complex to generalize well

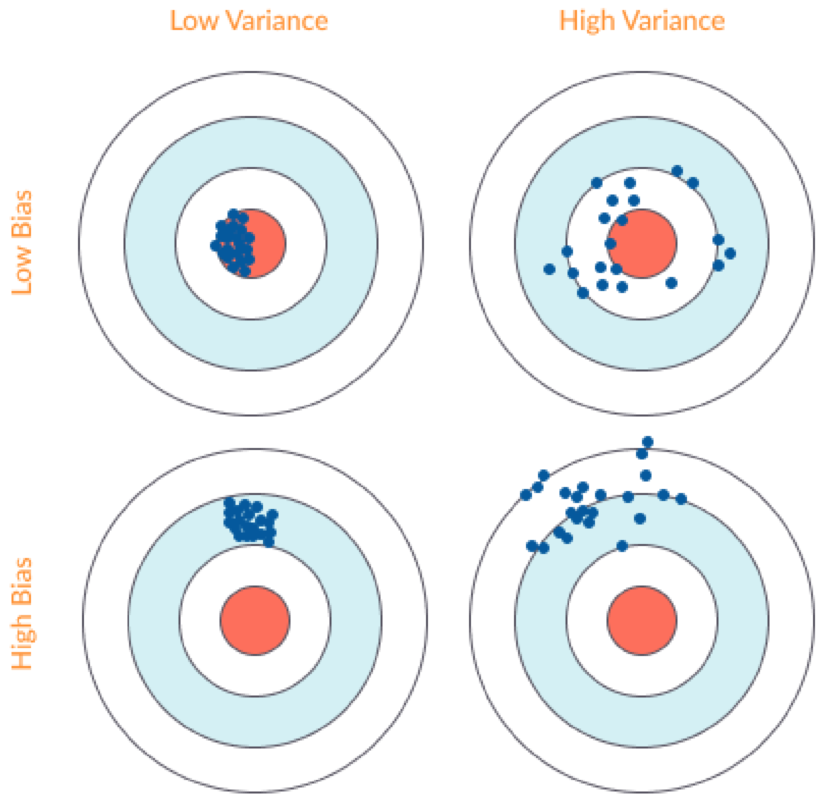
⇔ validation error large

Low variance ⇔ model simple enough to generalize well

SOURCE: HTTPS://TOWARDSDATASCIENCE.COM/UNDERSTANDING-THE-BIAS-VARIANCE-TRADEOFF-165E6942B229

Bias-Variance

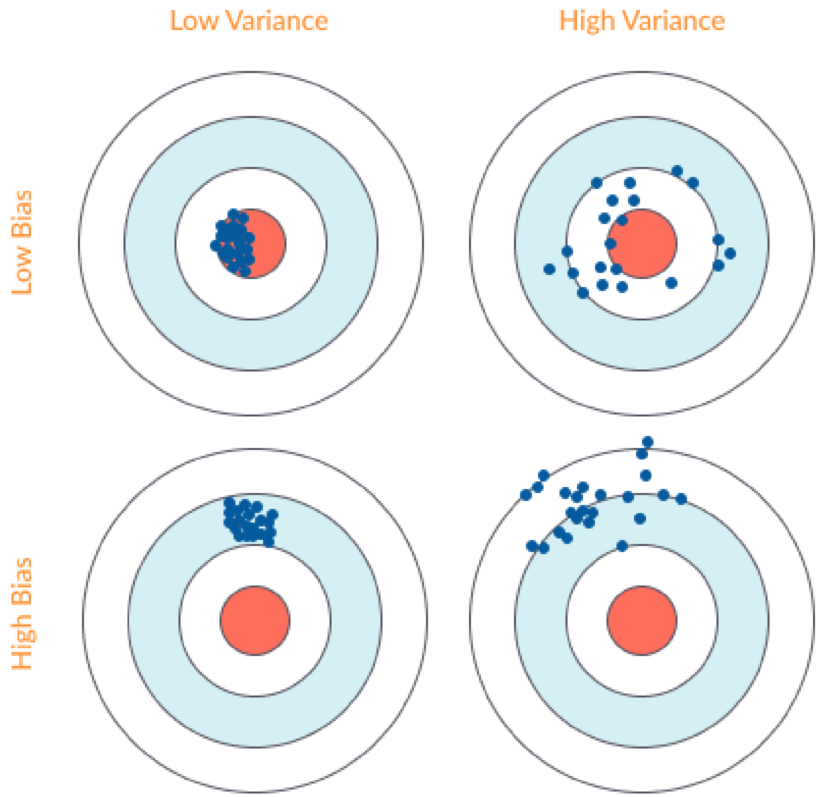
Repeat the model building process multiple times



The centre of the target is a model that perfectly predicts the correct values

Bias-Variance

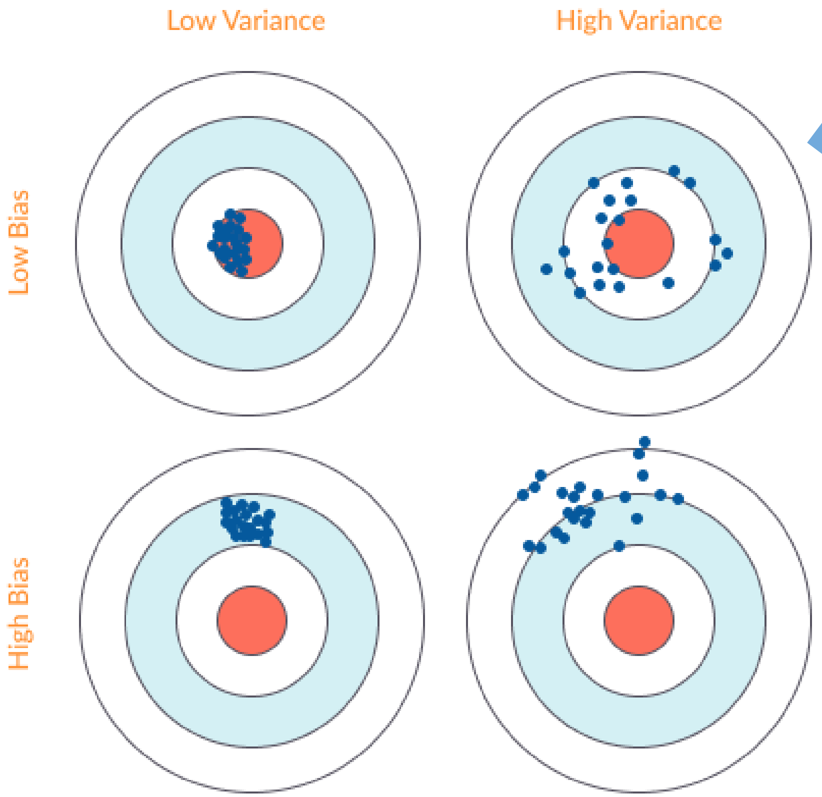
Repeat the model building process multiple times



underfitting

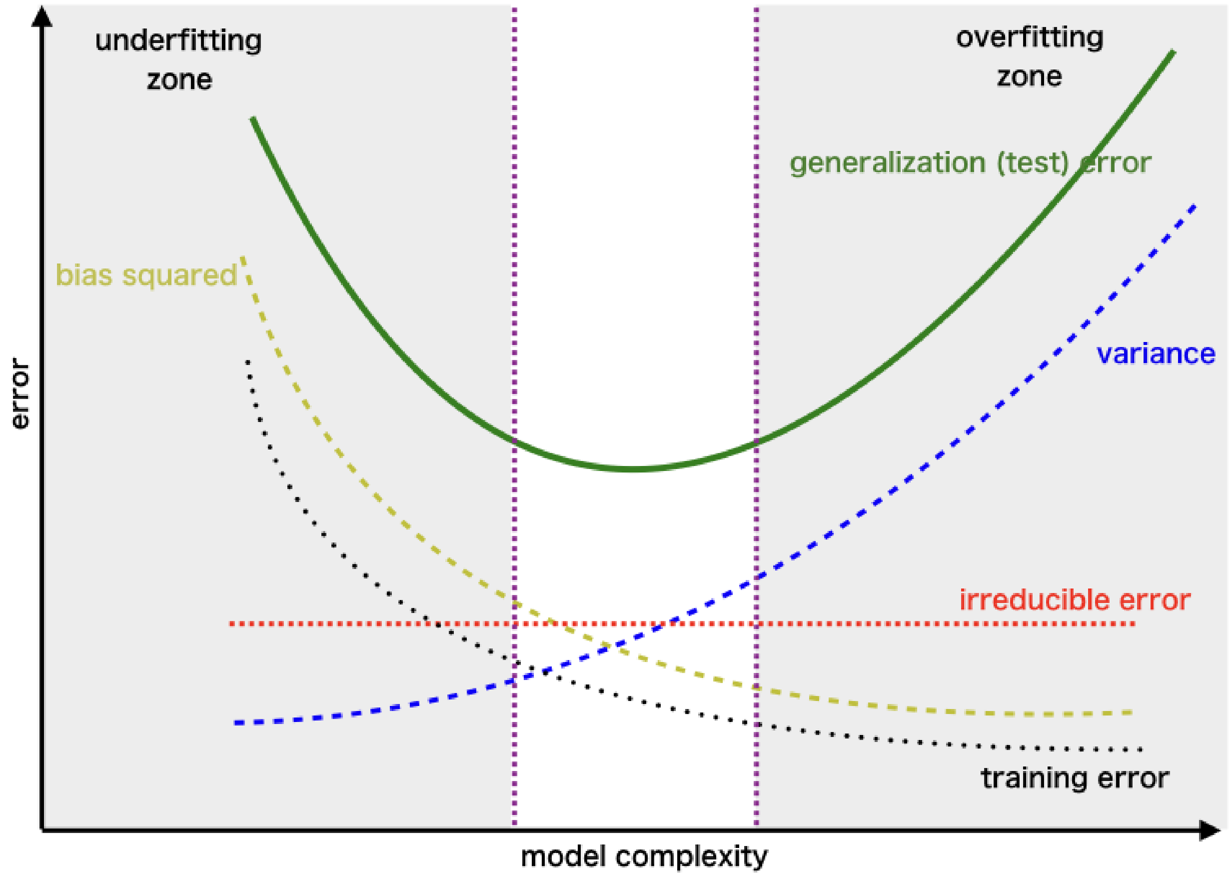
Bias-Variance

Repeat the model building process multiple times



overfitting

Bias-Variance Trade-Off



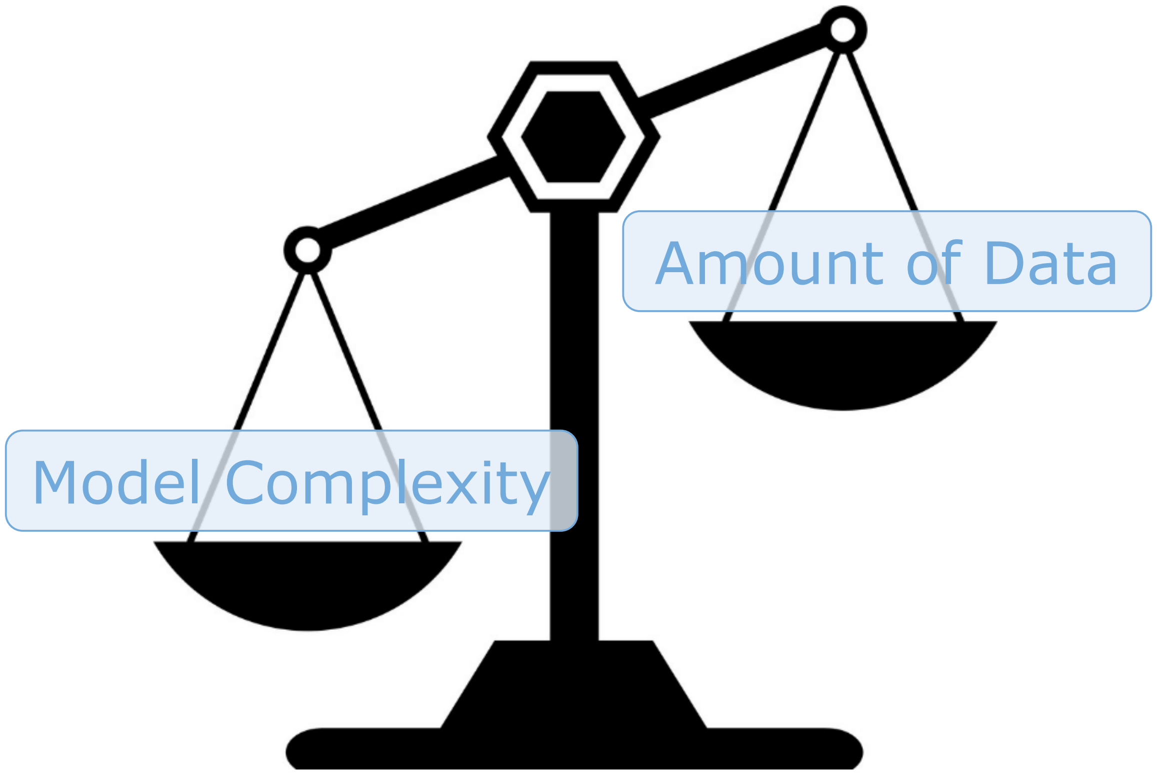
Issues

- Deep neural networks are prone to overfitting (highly complex models)
- Model complexity tied to task complexity

Examples of Competitive DNN

	VGGNet	DeepVideo	GNMT
Used For	Identifying Image Category	Identifying Video Category	Translation
Input	Image	Video	English Text
Output	1000 Categories	47 Categories	French Text
Parameters	140M	~100M	380M
Data Size	1.2M Images with assigned Category	1.1M Videos with assigned Category	6M Sentence Pairs, 340M Words
Dataset	ILSVRC-2012	Sports-1M	WMT'14
	2014	2014	2016

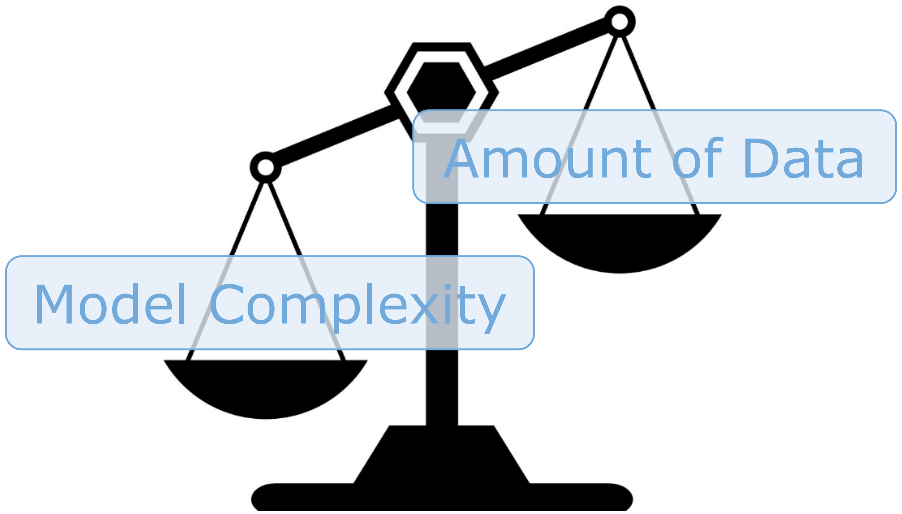
Need to find balance



What if limited data?

- Expensive, labor-intensive to collect
- Usage restriction
  - Sensitive data (confidentiality issues)
- Class imbalance
  - More healthy people than people with a given disease

What if limited data?



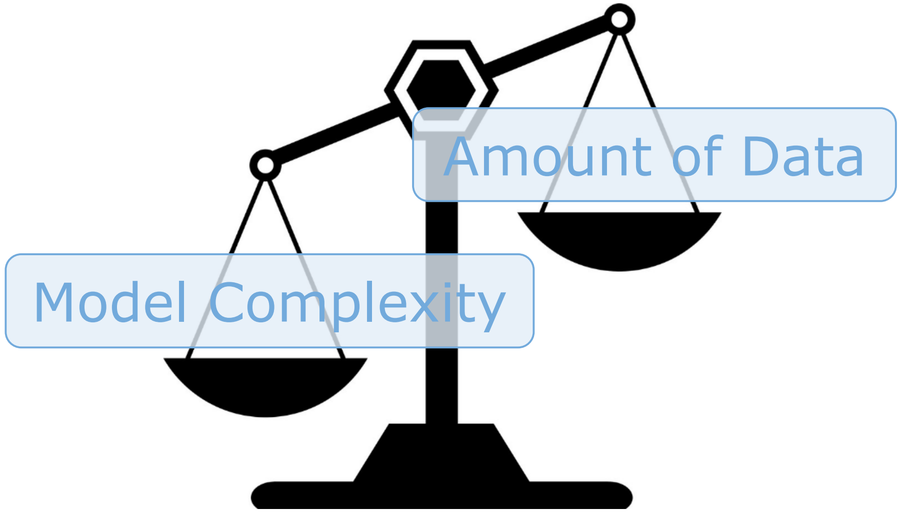
What if limited data?

Reduce Model Complexity

L2 regularisation

L1 regularisation

Dropout



L2 regularisation (weight decay)

- Idea: constrain the network weights by adding a regularization term to the loss function $J(\boldsymbol{w})$

L2 regularisation (weight decay)

- Idea: constrain the network weights by adding a regularization term to the loss function $J(\boldsymbol{w})$

$$\underbrace{\tilde{J}(\boldsymbol{w})}_{\text{New loss function}} = J(\boldsymbol{w}) + \alpha \underbrace{||\boldsymbol{w}||_2^2}_{\text{Regularization term}}$$

L2 regularisation: gradient

$$\tilde{J}(\boldsymbol{w}) = J(\boldsymbol{w}) + \alpha \, \|\boldsymbol{w}\|_2^2$$

- Gradient update:

$$\nabla_{\boldsymbol{w}} \tilde{J}(\boldsymbol{w}) = \alpha \boldsymbol{w} + \nabla_{\boldsymbol{w}} J(\boldsymbol{w})$$

$$\boldsymbol{w}_{\text{new}} = (1 - \eta\alpha)\boldsymbol{w}_{old} - \eta \nabla_{\boldsymbol{w}} J(\boldsymbol{w}_{old})$$

L1 regularisation (LASSO)

- Idea: constrain the network weights by adding a regularization term to the loss function $J(\boldsymbol{w})$

L1 regularisation (LASSO)

- Idea: constrain the network weights by adding a regularization term to the loss function $J(\boldsymbol{w})$

$$\underbrace{\tilde{J}(\boldsymbol{w})}_{\text{New loss function}} = J(\boldsymbol{w}) + \alpha \underbrace{||\boldsymbol{w}||_1}_{\text{Regularization term}}$$

L1 regularisation: gradient

$$\tilde{J}(\boldsymbol{w}) = J(\boldsymbol{w}) + \alpha \|\boldsymbol{w}\|_1$$

- Gradient:

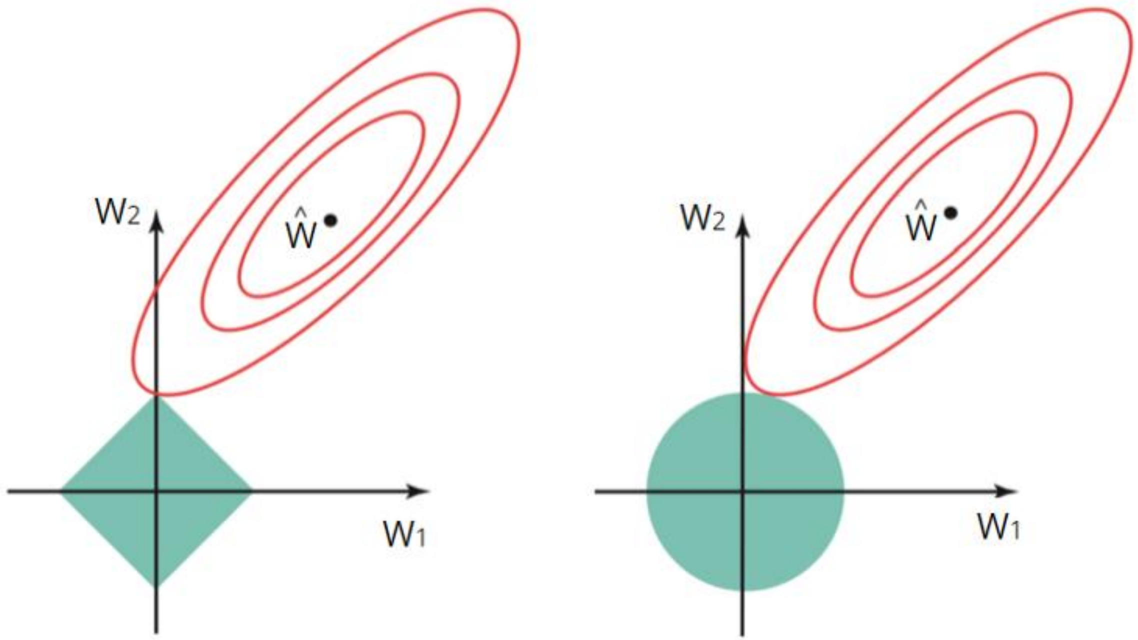
$$\nabla_{\boldsymbol{w}} \tilde{J}(\boldsymbol{w}) = \alpha \operatorname{sign}(\boldsymbol{w}) + \nabla_{\boldsymbol{w}} J(\boldsymbol{w})$$

How does regularization work?

- Main idea: smaller weights reduce the impact of the hidden neurons, they become neglectable and the overall complexity of the neural network gets reduced.

How does regularization work?

2D example



L1 regularisation

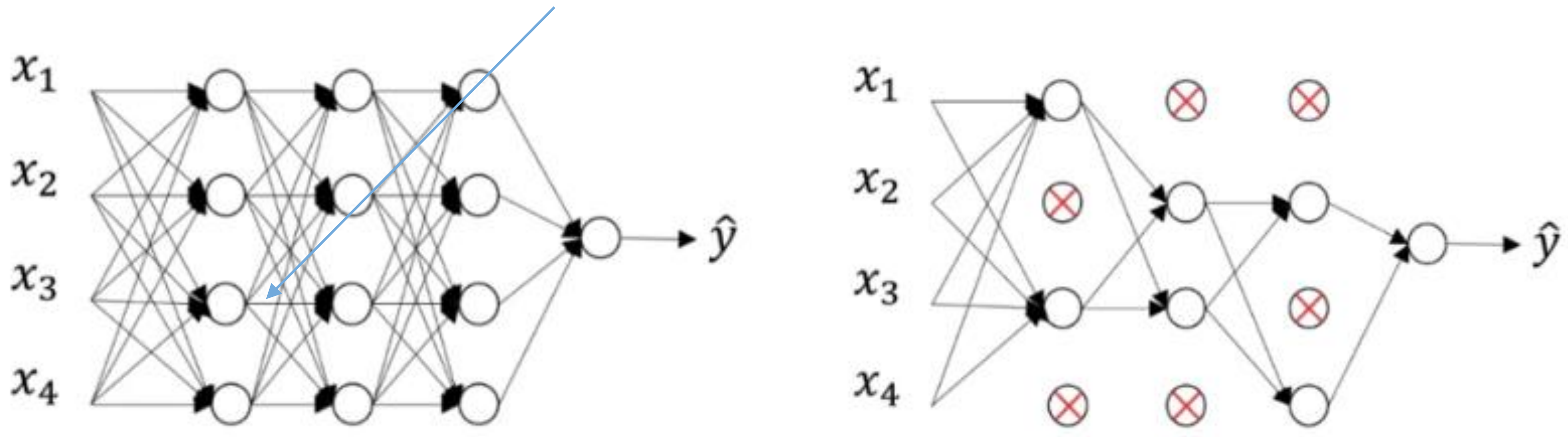
L2 regularisation

Contours of loss function

Constraint function

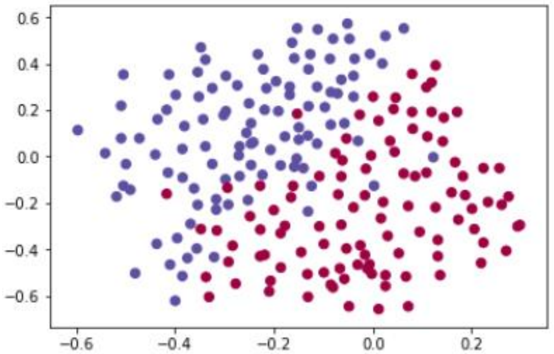
Dropout

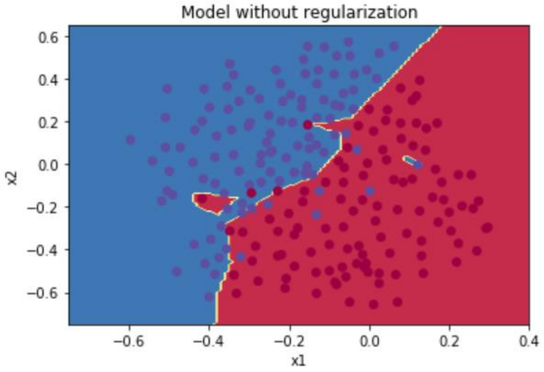
- During training turn off a neuron with some probability p

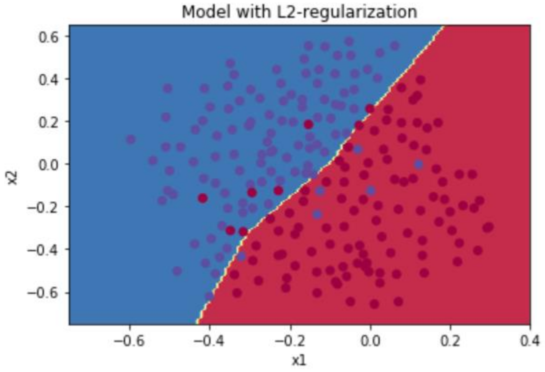


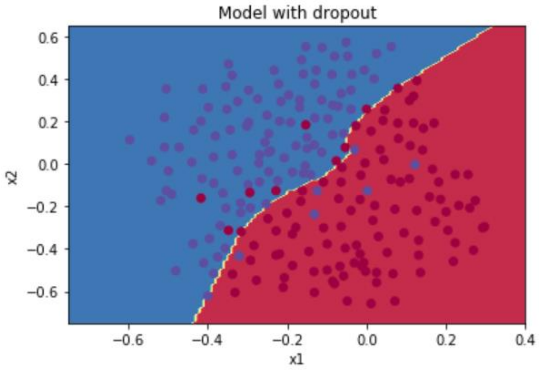
- Idea: The NN will be reluctant to give high weights to certain features, because they might disappear ➔ weights spread across all features making them smaller

Regularization Example









On the training set:
Accuracy: 0.94786729
On the test set:
Accuracy: 0.915

On the train set:
Accuracy: 0.938388
On the test set:
Accuracy: 0.93

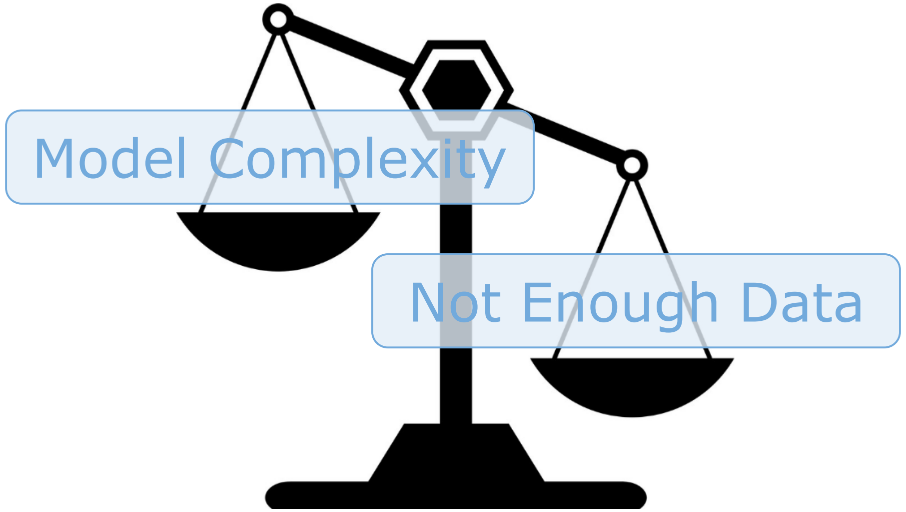
On the train set:
Accuracy: 0.928909
On the test set:
Accuracy: 0.95

SOURCE: HTTPS://TOWARDSDATASCIENCE.COM/HOW-TO-IMPROVE-A-NEURAL-NETWORK-WITH-REGULARIZATION-8A18ECDA9FE3

What if limited data?

Increase Amount of Data

Data Augmentation



Data Augmentation

- Idea: generate synthetic data from the training data

Transformative Approaches | Generative Approaches

- The new data must *preserve* the label or the label must be modified accordingly

Data Augmentation Overall

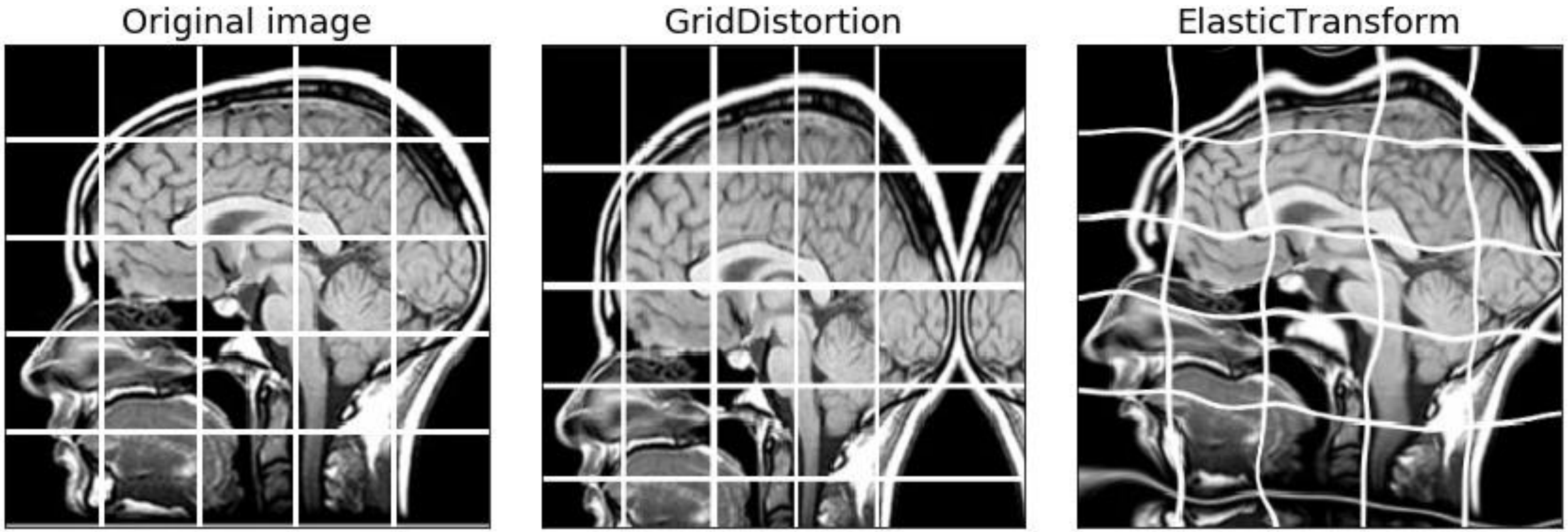
- Increase size and diversity of training data
- Learn invariance to some transformations
- Implicit regularisation effects
- Noising ⇔ data augmentation

Data Augmentation for Images

SOURCE: BUSLAEV, A., IGLOVIKOV, V. I., KHVEDCHENYA, E., PARINOV, A., DRUZHININ, M., & KALININ, A. A. (2020). ALBUMENTATIONS: FAST AND FLEXIBLE IMAGE AUGMENTATIONS. *INFORMATION*, *11*(2), 125.

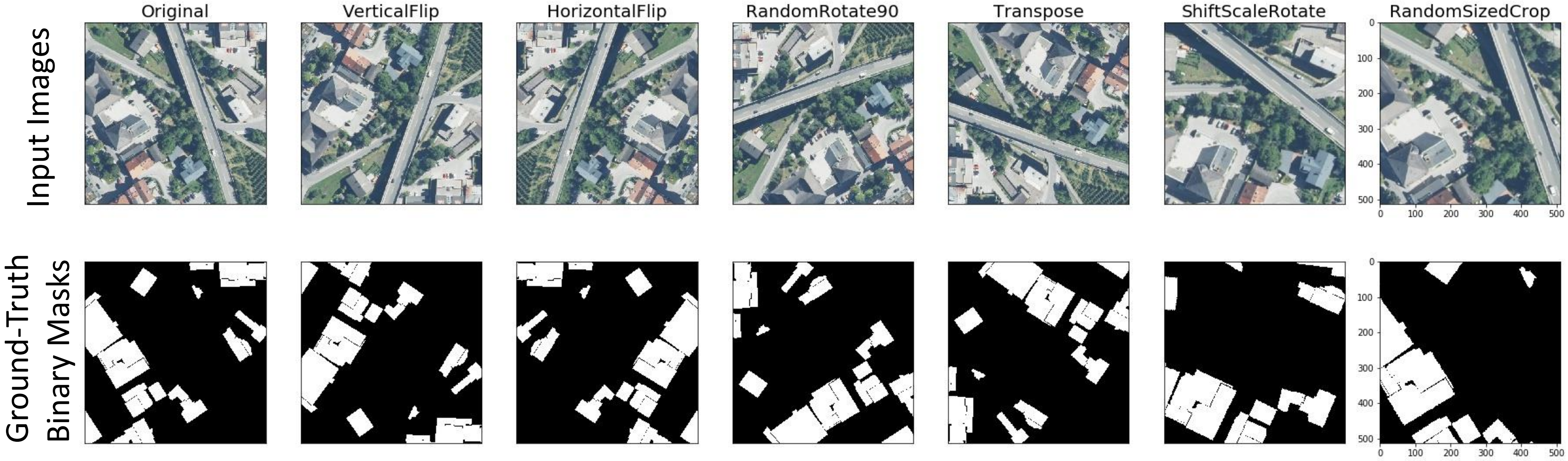
Data Augmentation for Images

- Example of grid transformations commonly used in biomedical image analysis



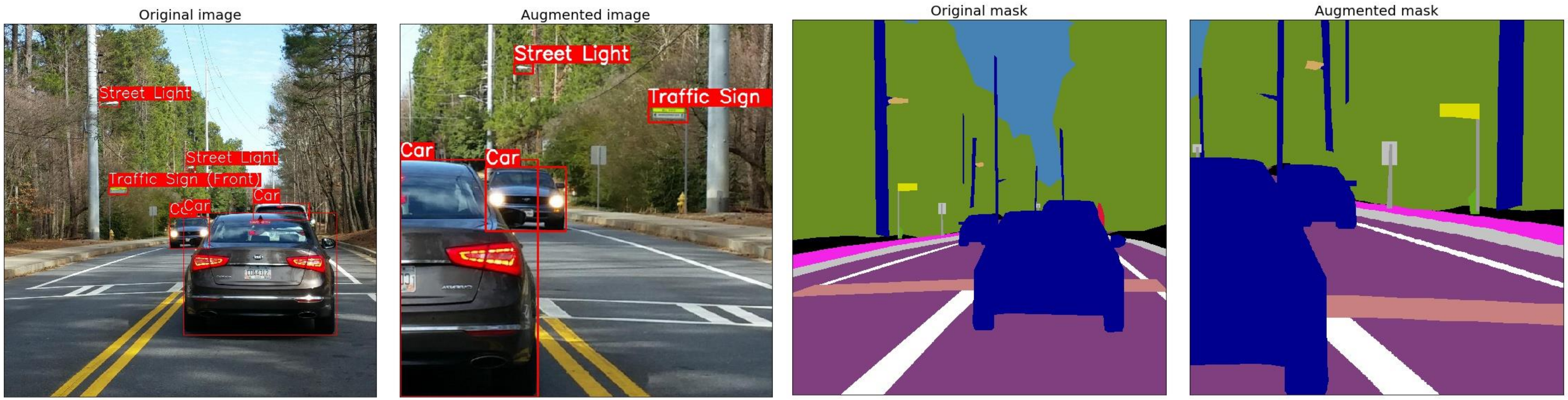
Data Augmentation for Images

- Example of geometry-preserving transforms in a segmentation task



Data Augmentation for Images

- Multiple targets task: an example of applying a combination of transformations to the original image, bounding boxes, and ground truth masks for instance segmentation



SOURCE: BUSLAEV, A., IGLOVIKOV, V. I., KHVEDCHENYA, E., PARINOV, A., DRUZHININ, M., & KALININ, A. A. (2020). ALBUMENTATIONS: FAST AND FLEXIBLE IMAGE AUGMENTATIONS. *INFORMATION*, *11*(2), 125.

Data Augmentation for Images

- Example of results for image classification

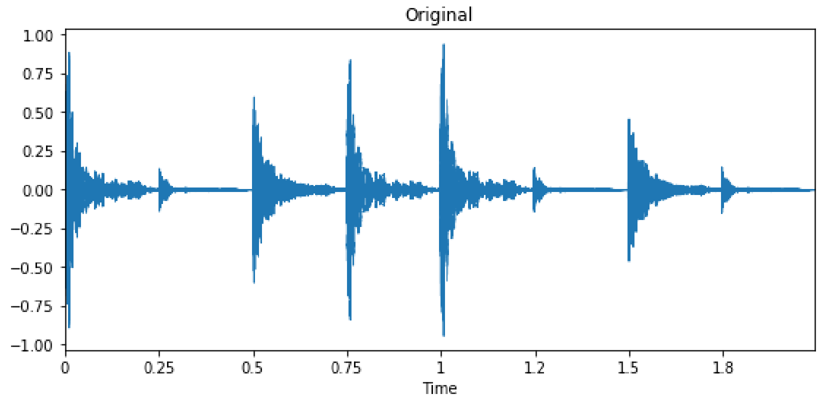
Method	C10	C10+	C100	C100+
ResNet18 [5]	10.63 ± 0.26	4.72 ± 0.21	36.68 ± 0.57	22.46 ± 0.31
ResNet18 + cutout	9.31 ± 0.18	3.99 ± 0.13	34.98 ± 0.29	21.96 ± 0.24
WideResNet [22]	6.97 ± 0.22	3.87 ± 0.08	26.06 ± 0.22	18.8 ± 0.08
WideResNet + cutout	$\mathbf{5.54 \pm 0.08}$	3.08 ± 0.16	$\mathbf{23.94 \pm 0.15}$	18.41 ± 0.27
Shake-shake regularization [4]	-	2.86	-	15.85
Shake-shake regularization + cutout	-	$\mathbf{2.56 \pm 0.07}$	-	$\mathbf{15.20 \pm 0.21}$

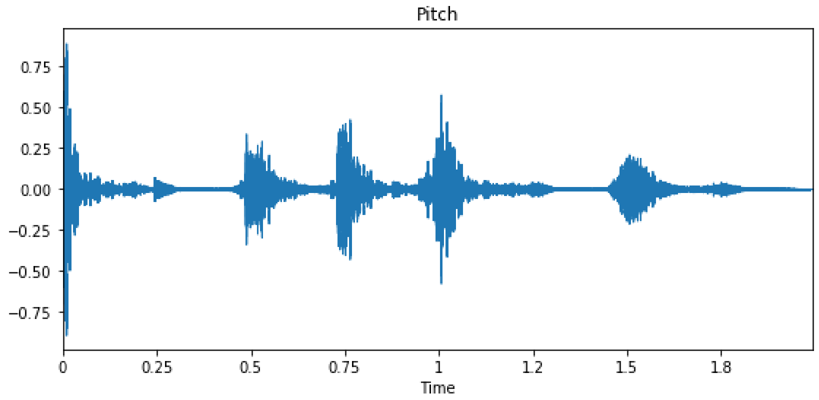
- Current works focus on *automatically* learning augmentation *schedules*

SOURCE: DEVRIES, T., & TAYLOR, G. W. (2017). IMPROVED REGULARIZATION OF CONVOLUTIONAL NEURAL NETWORKS WITH CUTOUT. *ARXIV PREPRINT ARXIV:1708.04552*.

Data Augmentation for Audio

- Noise Injection
- Time shifting
- Pitch change
- Speed change
- Background noise





Data Augmentation for NLP

"Everyone was in a good mood after enjoying delicious pizzas."

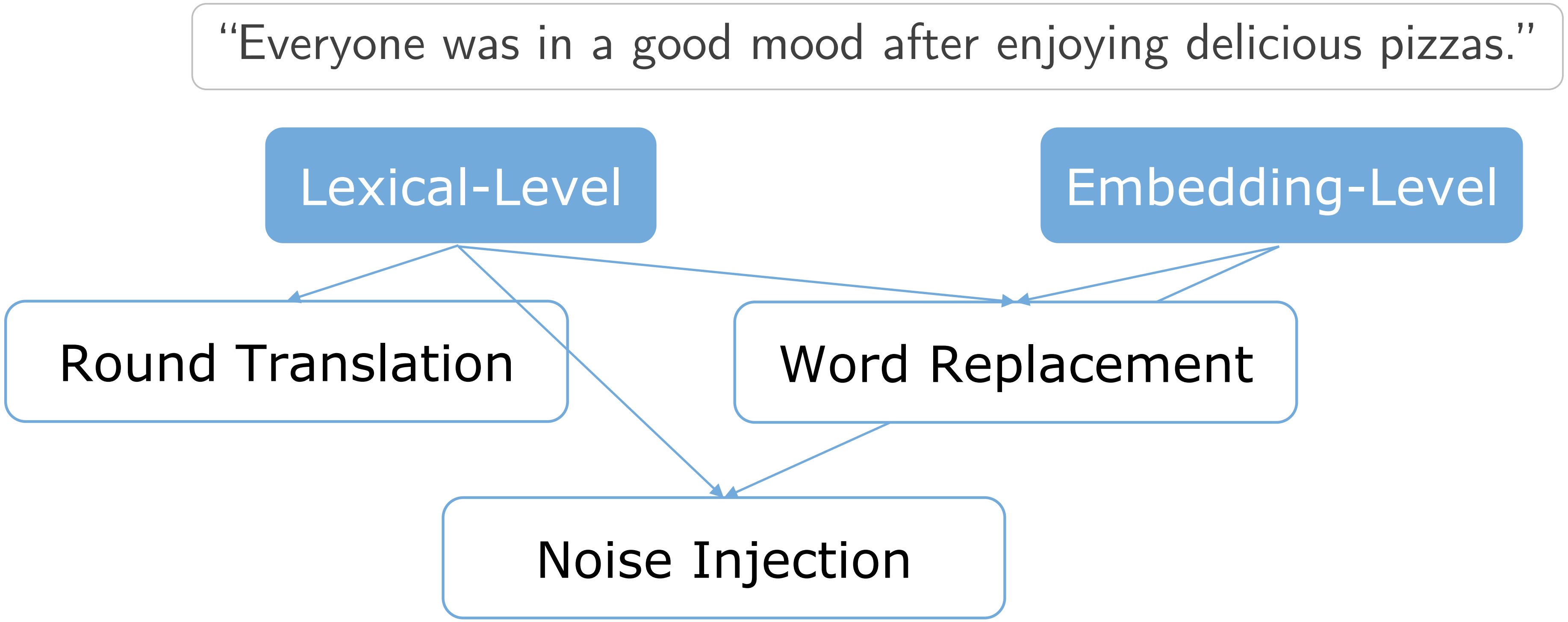
Data Augmentation for NLP

"Everyone was in a good mood after enjoying delicious pizzas."

Lexical-Level

Embedding-Level

Data Augmentation for NLP



Noise Injection

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Inserting, Deleting, Swapping random words

Noise Injection

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Inserting, Deleting, Swapping random words

"Everyone was in a good mood after enjoying delicious pizzas."

Noise Injection

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Inserting, Deleting, Swapping random words

"Everyone was in a good mood after enjoying task delicious pizzas."

Noise Injection

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Inserting, Deleting, Swapping random words

"Everyone was in a good ____ after enjoying task delicious pizzas."

Noise Injection

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Inserting, Deleting, Swapping random words

"Everyone good in a was _ _ _ _ _ after enjoying talk delicious pizzas."

Noise Injection

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Inserting, Deleting, Swapping random words

"Everyone good in a was after enjoying talk delicious pizzas."

- *Embedding-level:* Adding (e.g., Gaussian) noise to the embeddings

Noise Injection

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Inserting, Deleting, Swapping random words

"Everyone good in a was after enjoying talk delicious pizzas."

- *Embedding-level:* Adding (e.g., Gaussian) noise to the embeddings

Noise Injection

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Inserting, Deleting, Swapping random words

"Everyone good in a was after enjoying talk delicious pizzas."

- *Embedding-level:* Adding (e.g., Gaussian) noise to the embeddings

Word Replacement

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Replace with synonym, hypernym, language model, …

Word Replacement

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Replace with synonym, hypernym, language model, …

"Everyone was in a good mood after enjoying delicious pizzas."

Word Replacement

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Replace with synonym, hypernym, language model, …

"Everyone was in a cheerful mood after enjoying delicious pizzas."

Word Replacement

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Replace with synonym, hypernym, language model, …

"Everyone was in a good mood after enjoying delicious food."

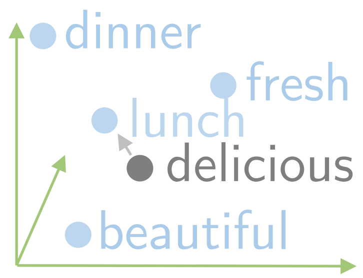
Word Replacement

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Replace with synonym, hypernym, language model, …

"Everyone was in a <u>cheerful</u> mood after enjoying delicious <u>food</u>."

- *Embedding-level:* Replace with nearest word embeddings



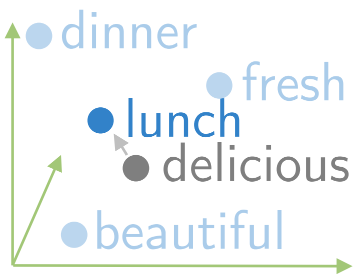
Word Replacement

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Replace with synonym, hypernym, language model, …

"Everyone was in a <u>cheerful</u> mood after enjoying delicious <u>food</u>."

- *Embedding-level:* Replace with nearest word embeddings



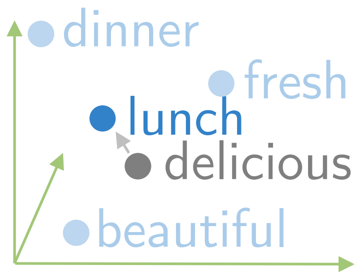
Word Replacement

"Everyone was in a good mood after enjoying delicious pizzas."

- *Lexical-level:* Replace with synonym, hypernym, language model, …

"Everyone was in a <u>cheerful</u> mood after enjoying delicious <u>food</u>."

- *Embedding-level:* Replace with nearest word embeddings



"Everyone was in a good mood after enjoying lunch pizzas."

Round-Translation

- Translation to a target language and then back to source language

"Everyone was in a good mood after enjoying delicious pizzas."

Round-Translation

- Translation to a target language and then back to source language

"Everyone was in a good mood after enjoying delicious pizzas."

En → Fr

«Tout le monde était de bonne humeur après avoir dégusté de délicieuses pizzas.»

Round-Translation

- Translation to a target language and then back to source language

"Everyone was in a good mood after enjoying delicious pizzas."

En → Fr

«Tout le monde était de bonne humeur après avoir dégusté de délicieuses pizzas.»

Fr → En

"Everyone was in a good mood after tasting delicious pizzas."

Data Augmentation for NLP

- Less widely used

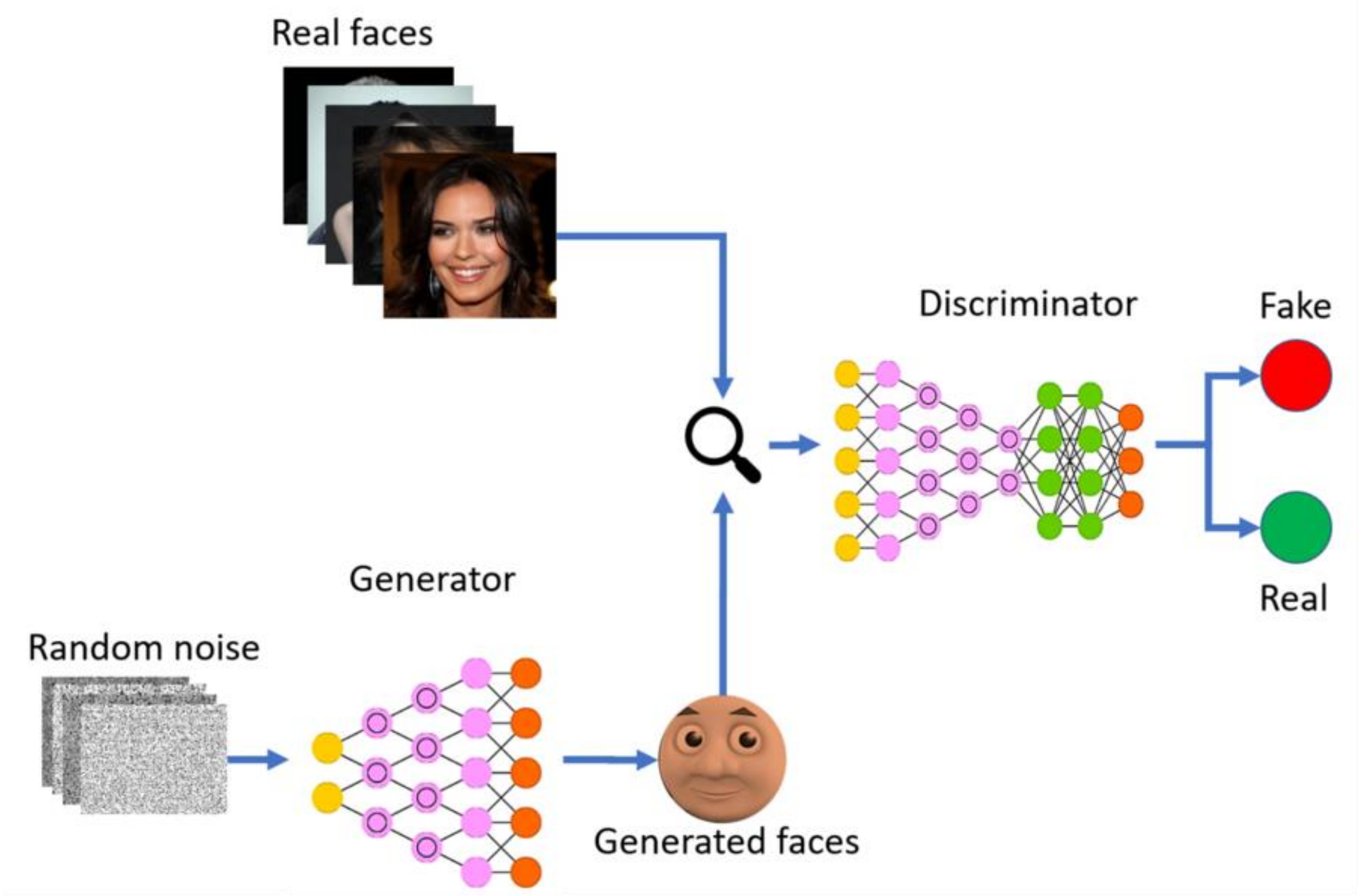
- Wider range of tasks with different invariance properties

Data Augmentation GANs

- Example for emotion classification
- 5%–10% increase in the classification accuracy

SOURCE: ZHU, X., LIU, Y., LI, J., WAN, T., & QIN, Z. (2018, JUNE). EMOTION CLASSIFICATION WITH DATA AUGMENTATION USING GENERATIVE ADVERSARIAL NETWORKS. IN PACIFIC-ASIA CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING (PP. 349-360). SPRINGER, CHAM.

GANs for faces generation



Data Augmentation for Big Data

- Can increase diversity

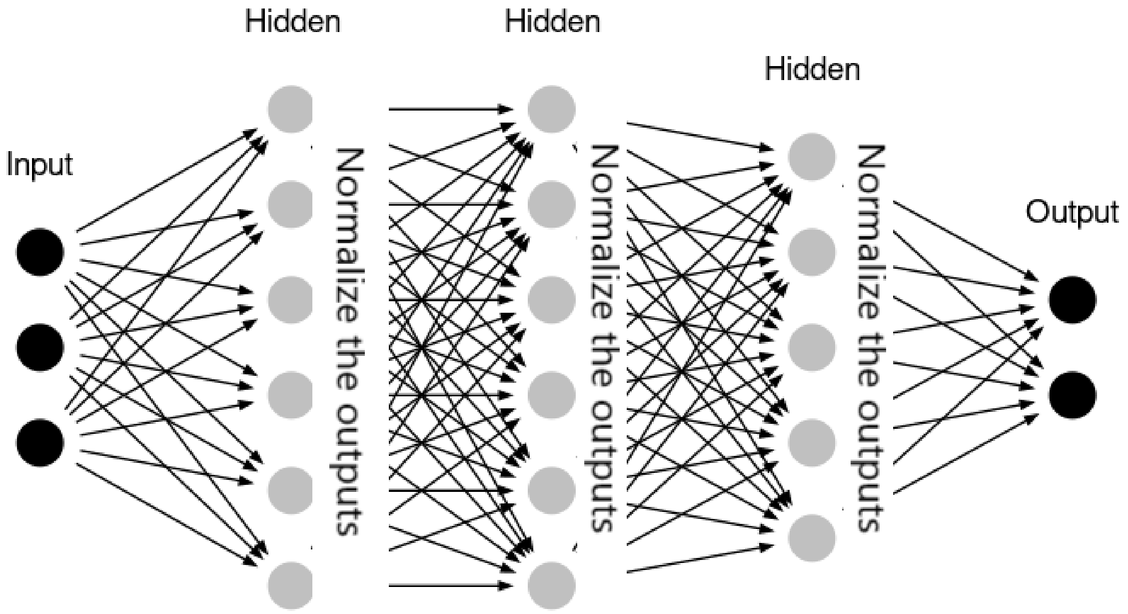
- Improve robustness

Be careful!

- Not every transformation ok!

Other tools

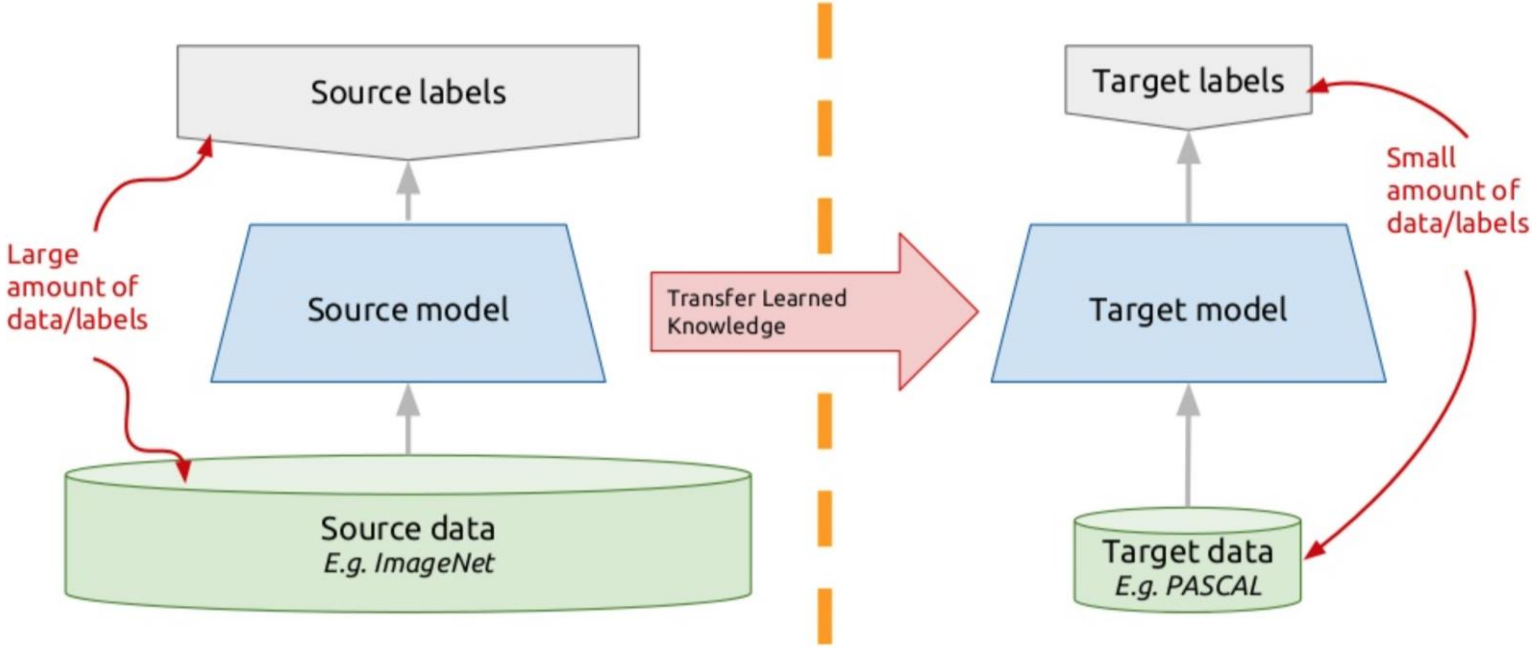
- Batch Normalization
  - Normalize layer inputs by subtracting the batch mean and dividing by the batch standard deviation



SOURCE: HTTPS://DEEPAI.ORG/MACHINE-LEARNING-GLOSSARY-AND-TERMS/BATCH-NORMALIZATION

Other tools

- Transfer Learning
  - Reuse parts of a previously trained model on a new network to solve a different but similar problem



SOURCE: HTTPS://MEDIUM.COM/THE-OFFICIAL-INTEGRATE-AI-BLOG/TRANSFER-LEARNING-EXPLAINED-7D275C1E34E2

References

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), 1-48.

https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

http://scott.fortmann-roe.com/docs/BiasVariance.html