Department of Medical Physics and Biomedical Engineering
Centre for Medical Image Computing (CMIC)
Wellcome / EPSRC Centre for Interventional and Surgical Sciences (WEISS)

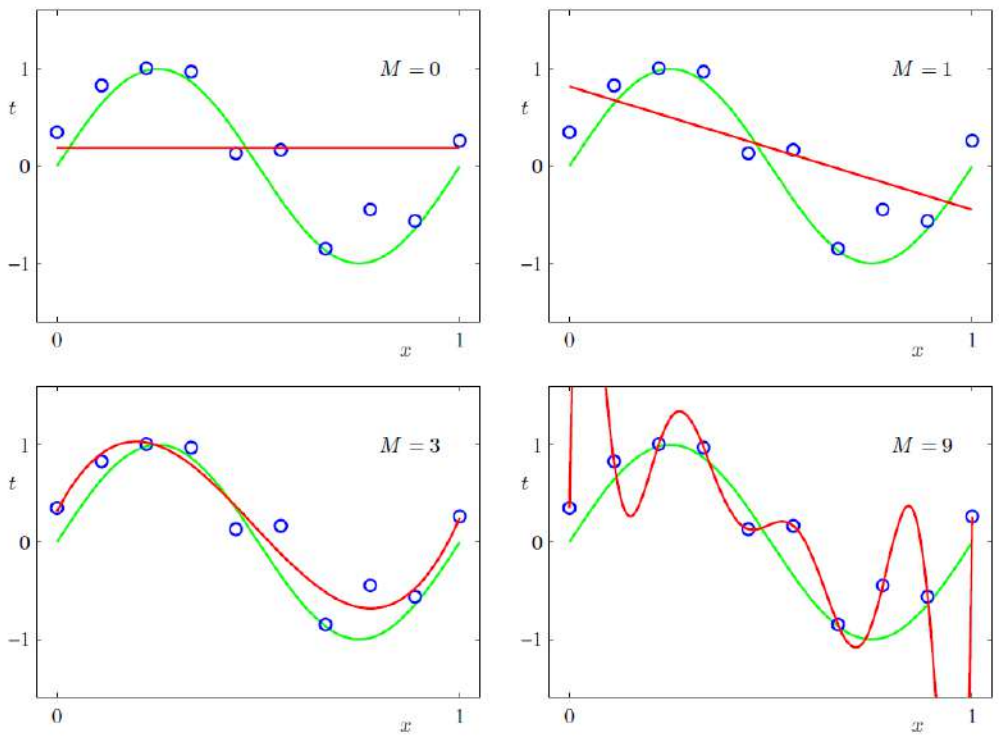
Deep Learning

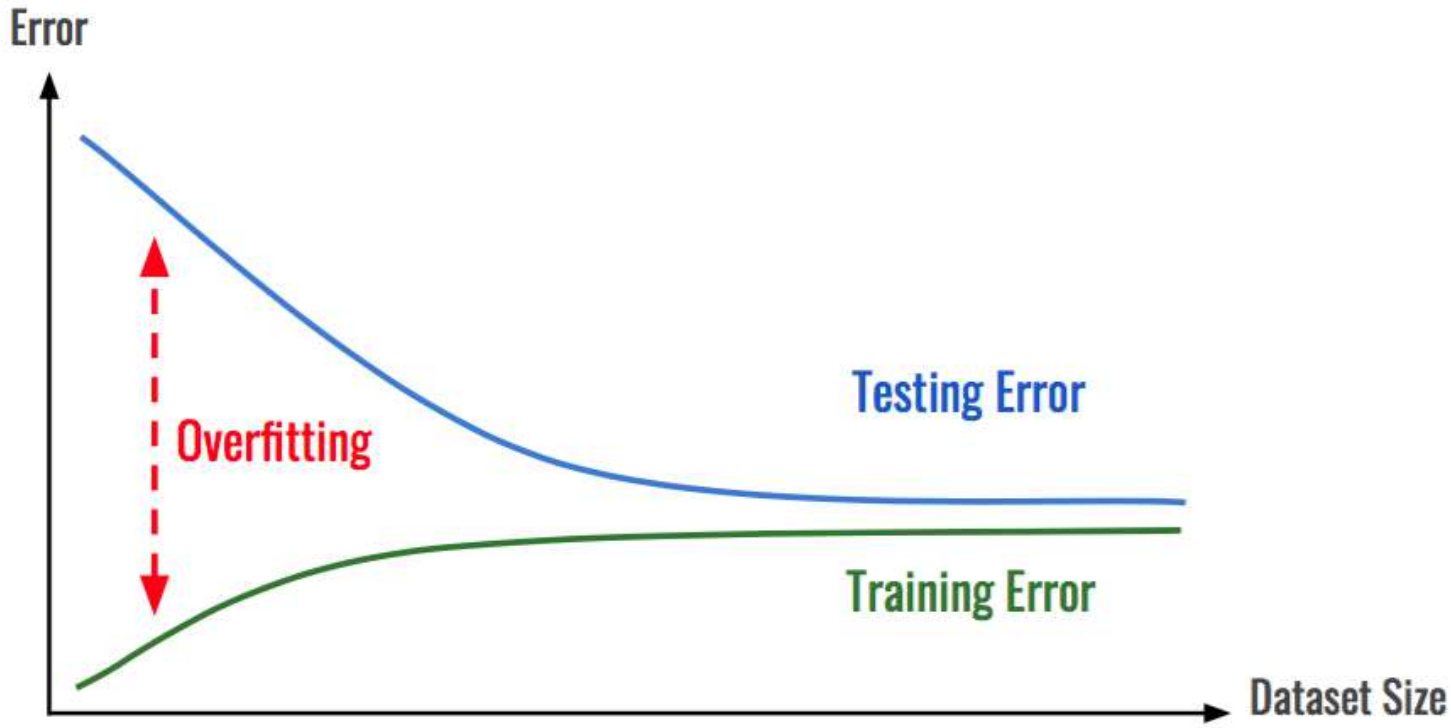
MPHY0041 Machine Learning in Medical Imaging

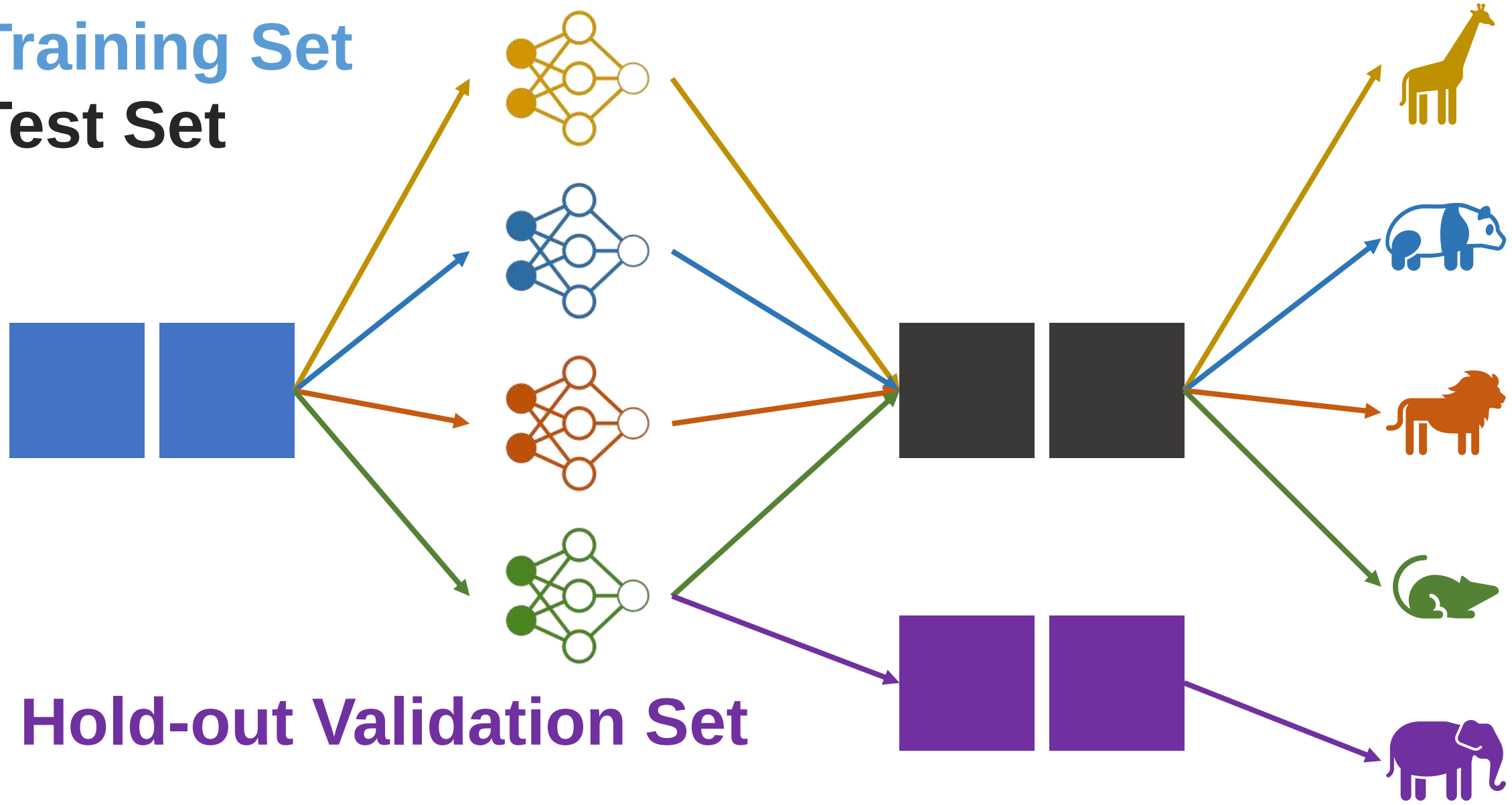
Yipeng Hu
yipeng.hu@ucl.ac.uk

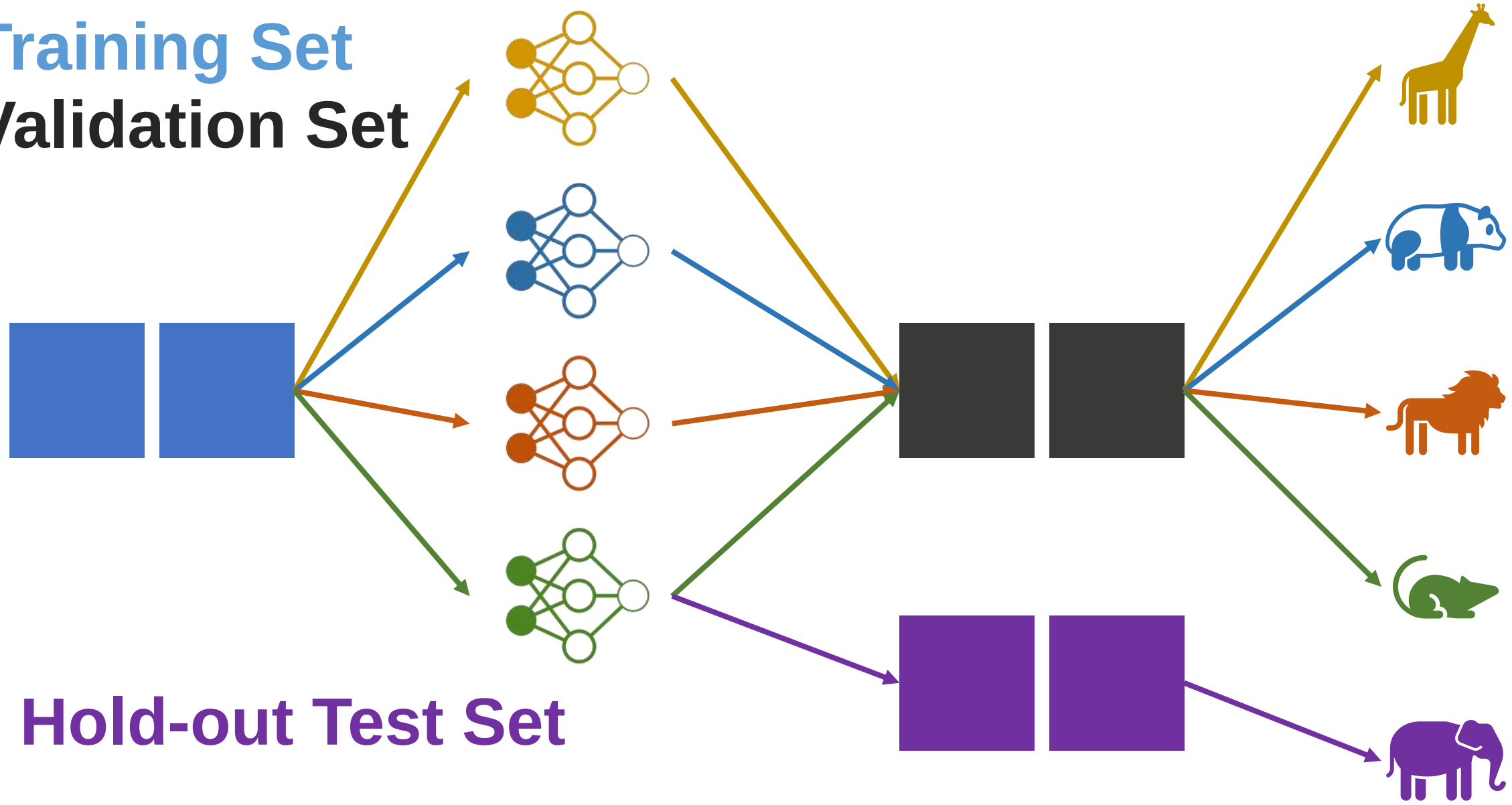
Training and Regularisation

Training and Regularisation






Training Set
Test Set
Hold-out Validation Set


Training Set
Validation Set
Hold-out Test Set

Training and Regularisation| Optimisation

Data generating distribution

$$J^*(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\mathrm{y})\sim p_{\mathrm{data}}} L(f(\boldsymbol{x};\boldsymbol{\theta}), y).$$

Training data distribution

$$J(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\mathrm{y})\sim \hat{p}_{\mathrm{data}}} L(f(\boldsymbol{x};\boldsymbol{\theta}), y),$$

Empirical risk

wrt. mini-batches

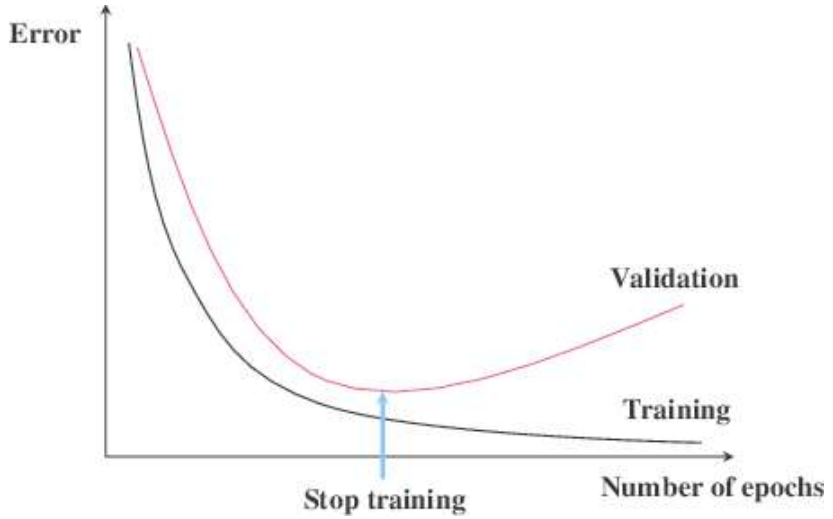
$$\mathbb{E}_{\boldsymbol{x},\mathrm{y}\sim\hat{p}_{\mathrm{data}}(\boldsymbol{x},y)}[L(f(\boldsymbol{x};\boldsymbol{\theta}), y)] = \frac{1}{m}\sum_{i=1}^{m} L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}), y^{(i)})$$

Large batches and small batches

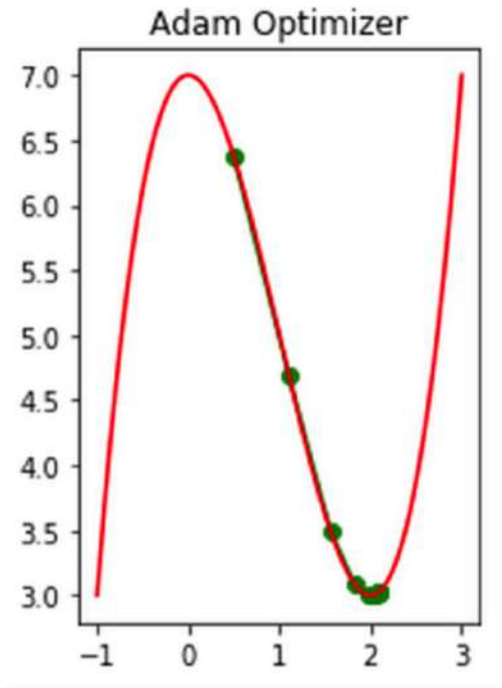
"Regulairsing for generalisability"

Surrogate losses

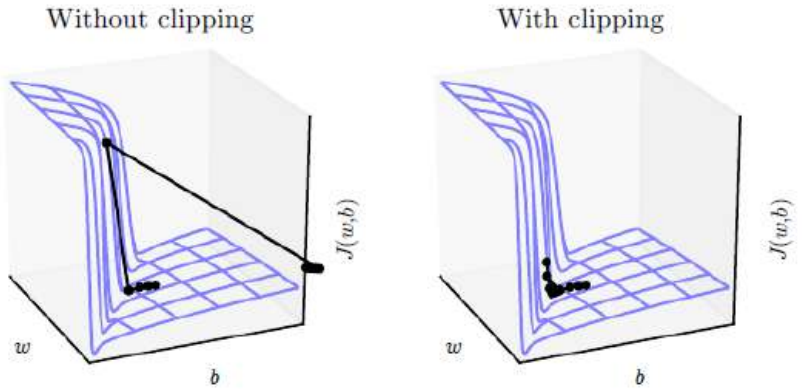
CE and Early stopping based on validation set



Local minima, non-identifiability and “a good enough solution”



Gradient explosion



Algorithms

- SGD and mini-batch gradient descent

$$\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^{m} L(\boldsymbol{f}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)}) \right)$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}.$$

- Momentum
- Parameter initialisation, wrt. Symmetry, depth etc
- Adaptive methods
  - high-order derivatives (Newton’s Method, Conjugate gradient)
  - Past derivative values (AdaGrad, RMSProp)
  - Adaptive momentum (Adam)

Reparameterisation for optimising deep networks

Batch normalisation and variants

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

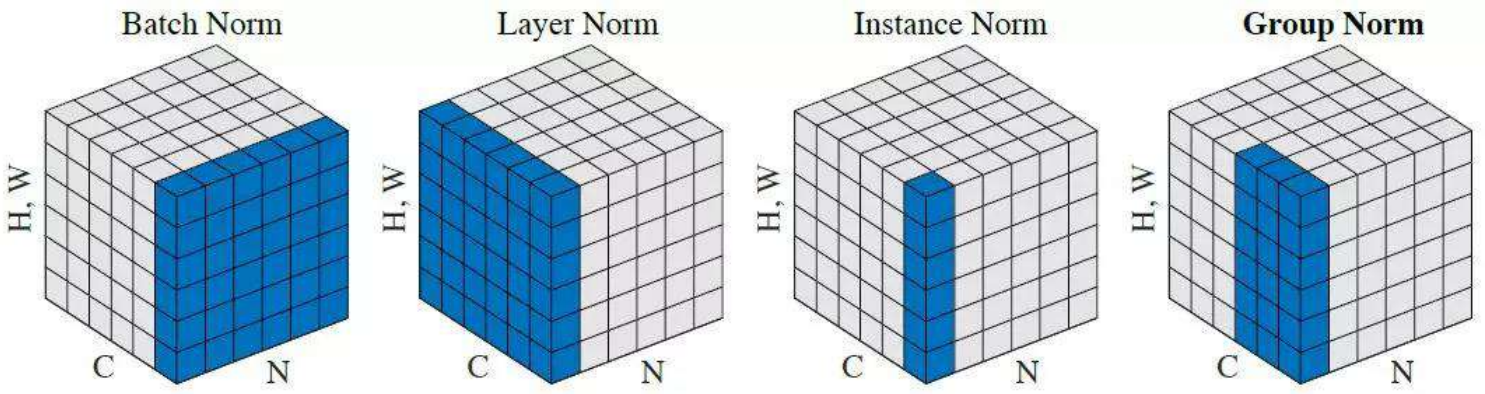
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

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

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

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

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.



Training and Regularisation| Constrain Parameters

Parameter norms

Weight decay by penalising L^2- and L^1-norms

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \text{const.}$$

Difference between L2- and L1-norms

$$\Omega(\boldsymbol{\theta}) = ||\boldsymbol{w}||_1 = \sum_i |w_i| \qquad \nabla_{\boldsymbol{w}}\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \alpha \text{sign}(\boldsymbol{w}) + \nabla_{\boldsymbol{w}} J(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{w})$$

L^2 norm gradient scales with w, therefore less likely to become zero, i.e. sparsity.

Parameter sharing

- CNN
- RNN
- Tiring: e.g.

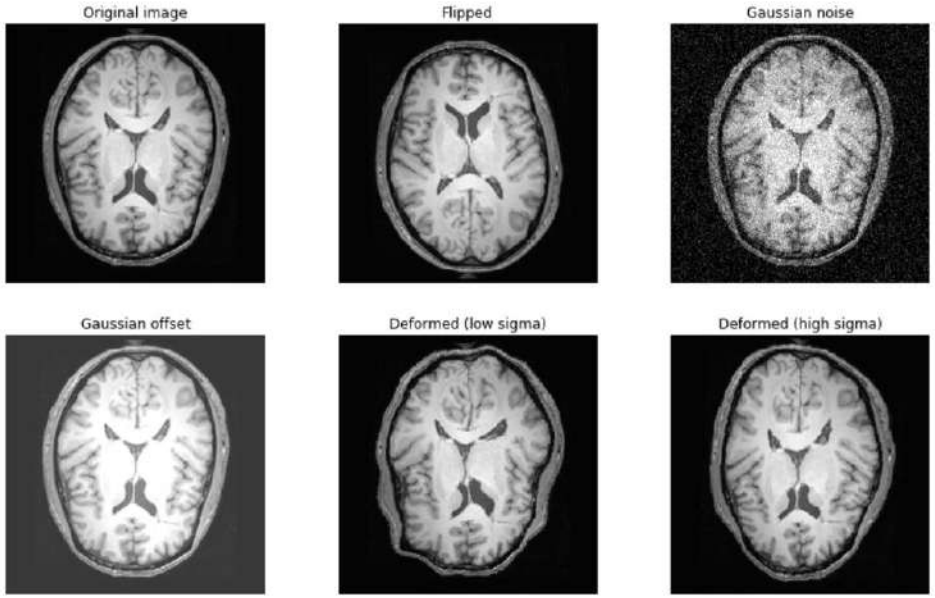
$$\Omega(\boldsymbol{w}^{(A)}, \boldsymbol{w}^{(B)}) = \|\boldsymbol{w}^{(A)} - \boldsymbol{w}^{(B)}\|_2^2$$

- Multi-task learning*
- Semi-supervised learning*

Training and Regularisation| Augment Data

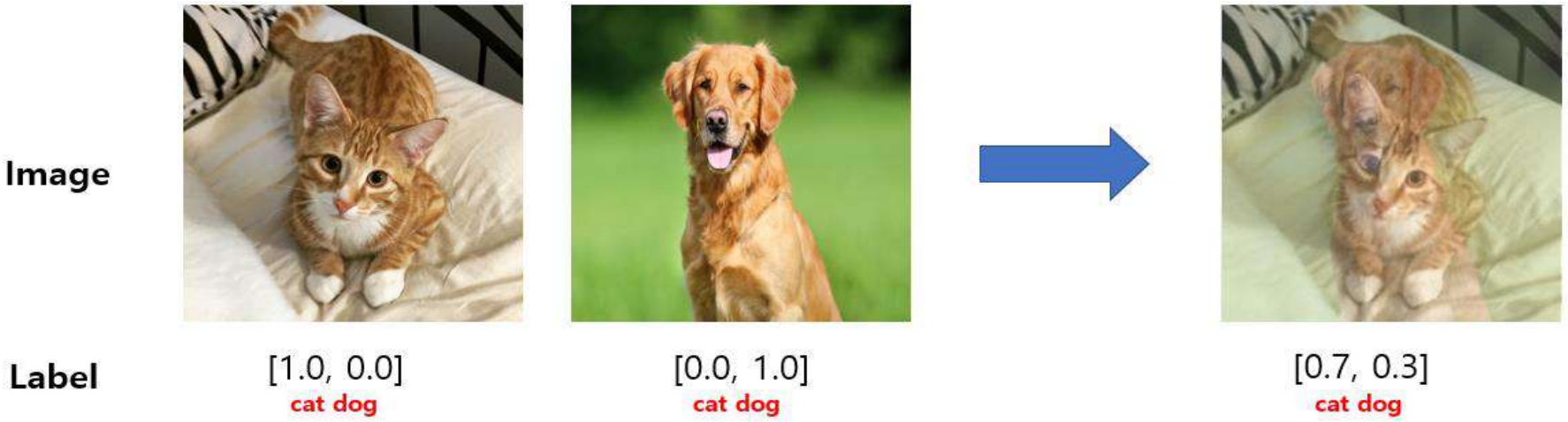
- Geometric transformation,
  - e.g. flipping, cropping, rotation
- Colour space,
  - e.g. RGB, intensity, contrast
- Imaging-specific,
  - e.g. photometric transformation for camera, speckle for ultrasound, bias field for MR, Projective transformation for CT
- Spatial transformation
  - inc. geometric transformation, affine, nonlinear deformation

- Generative models*



The “mixup”

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j, \quad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j, \quad \text{where } y_i, y_j \text{ are one-hot label encodings}$$



Training and Regularisation| Add Randomness

- Augment input*
- Add noise to weights
- Label smoothing (assuming noisy output)

- Dropout*

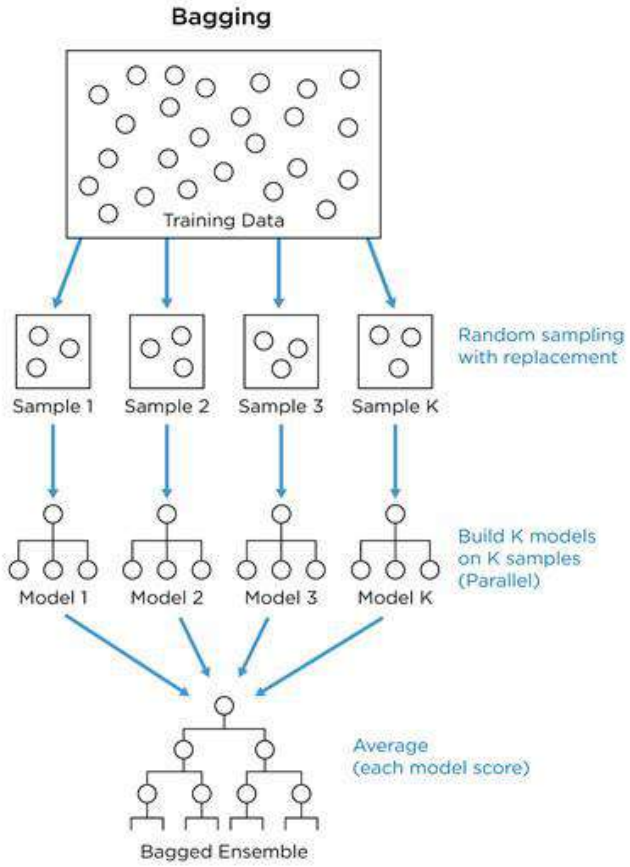
Training and Regularisation| Ensemble

Model Averaging

$$p(X) = \sum_{Y} p(X|Y)p(Y)$$

e.g. Bagging/bootstrap aggregating

- Weighted averaging
- Majority voting

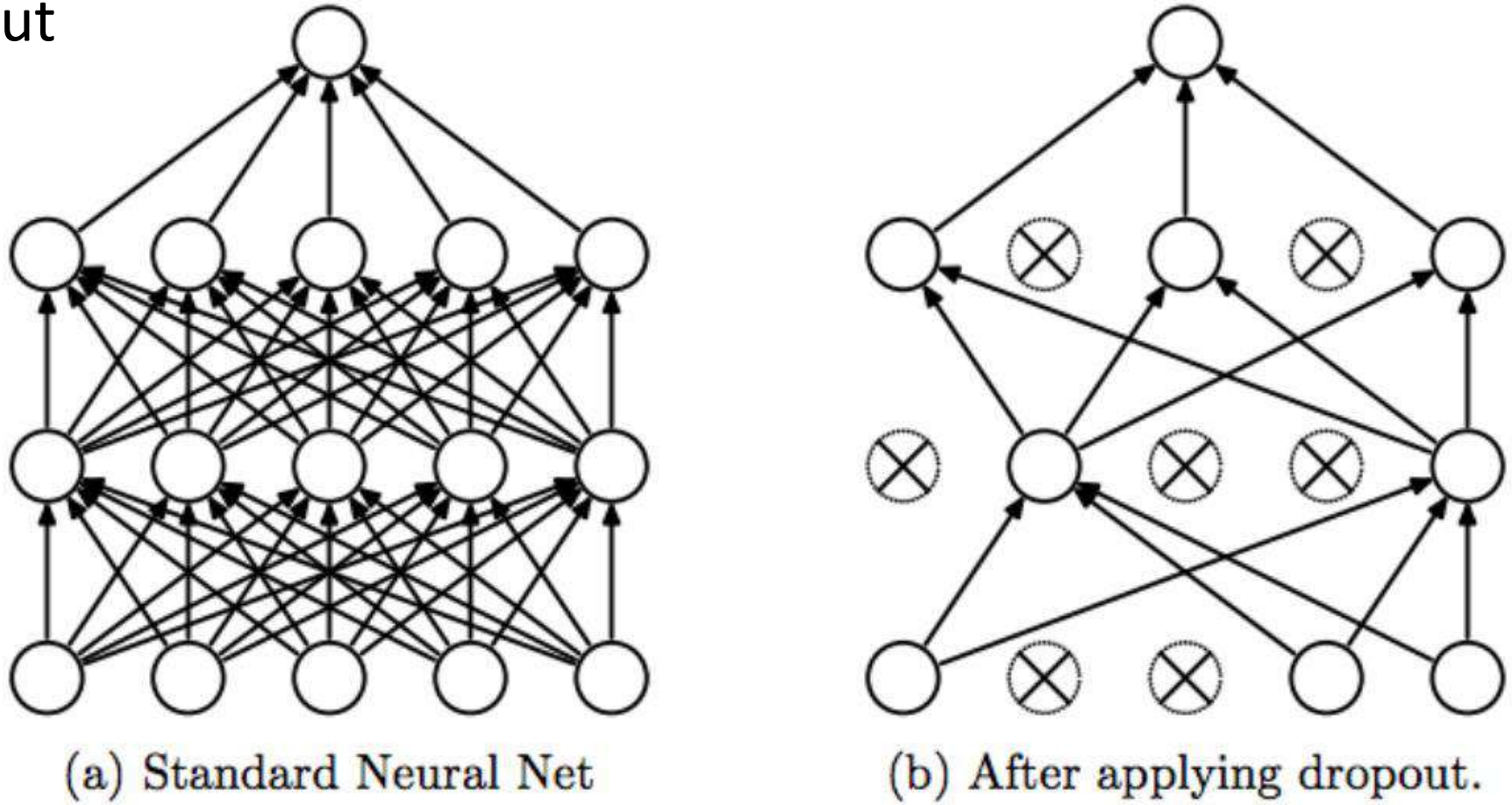


Boosting

Weak learners -> strong learners,
i.e. a meta-algorithm for reducing bias

Competition winners!

Dropout



(a) Standard Neural Net

(b) After applying dropout.

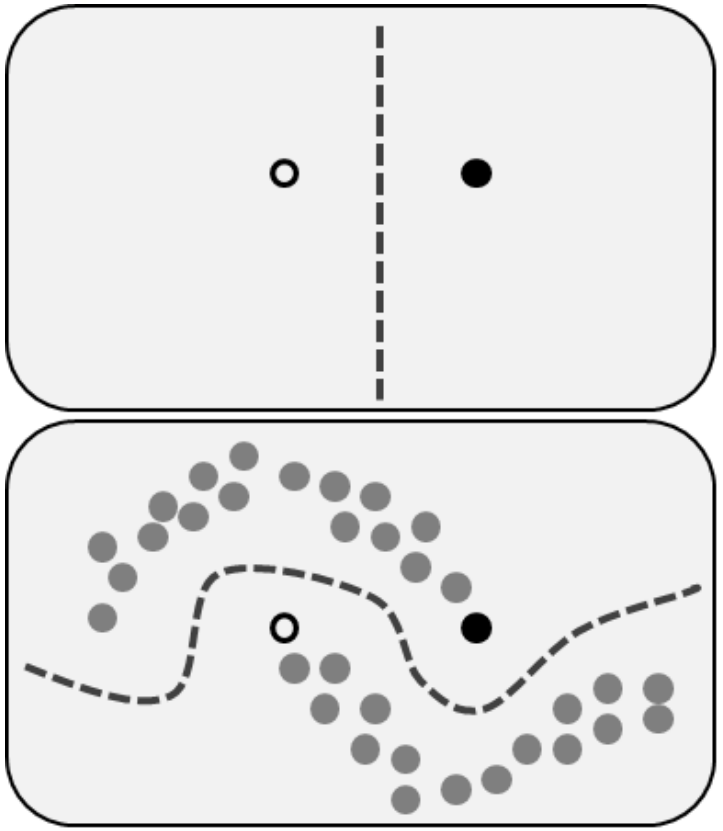
- *Test-time dropout for ensemble and uncertainty estimation*

Training and Regularisation| Semi-Supervised Learning

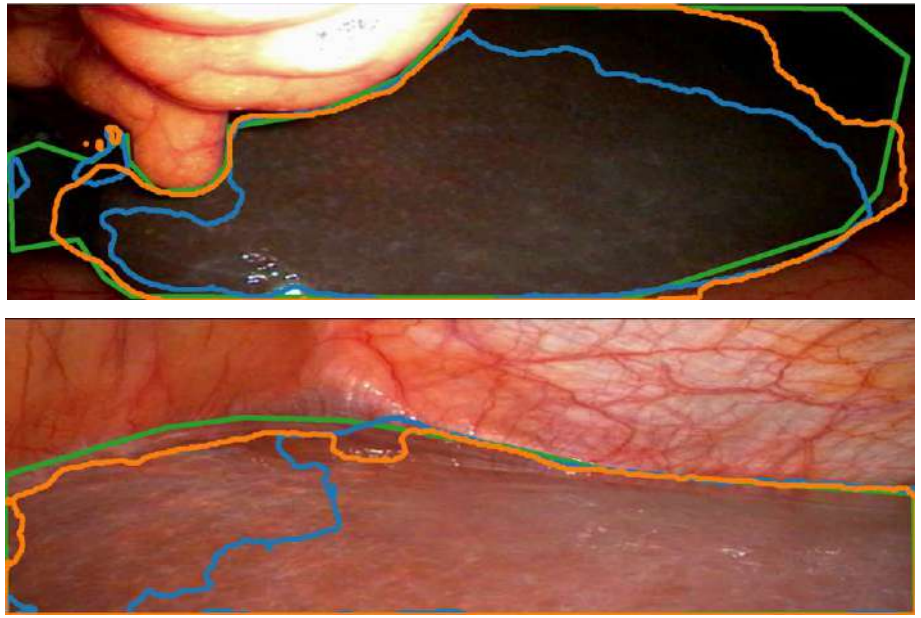
Supervised learning: labelled data = data + labels $P(\mathbf{x}, \mathbf{y})$

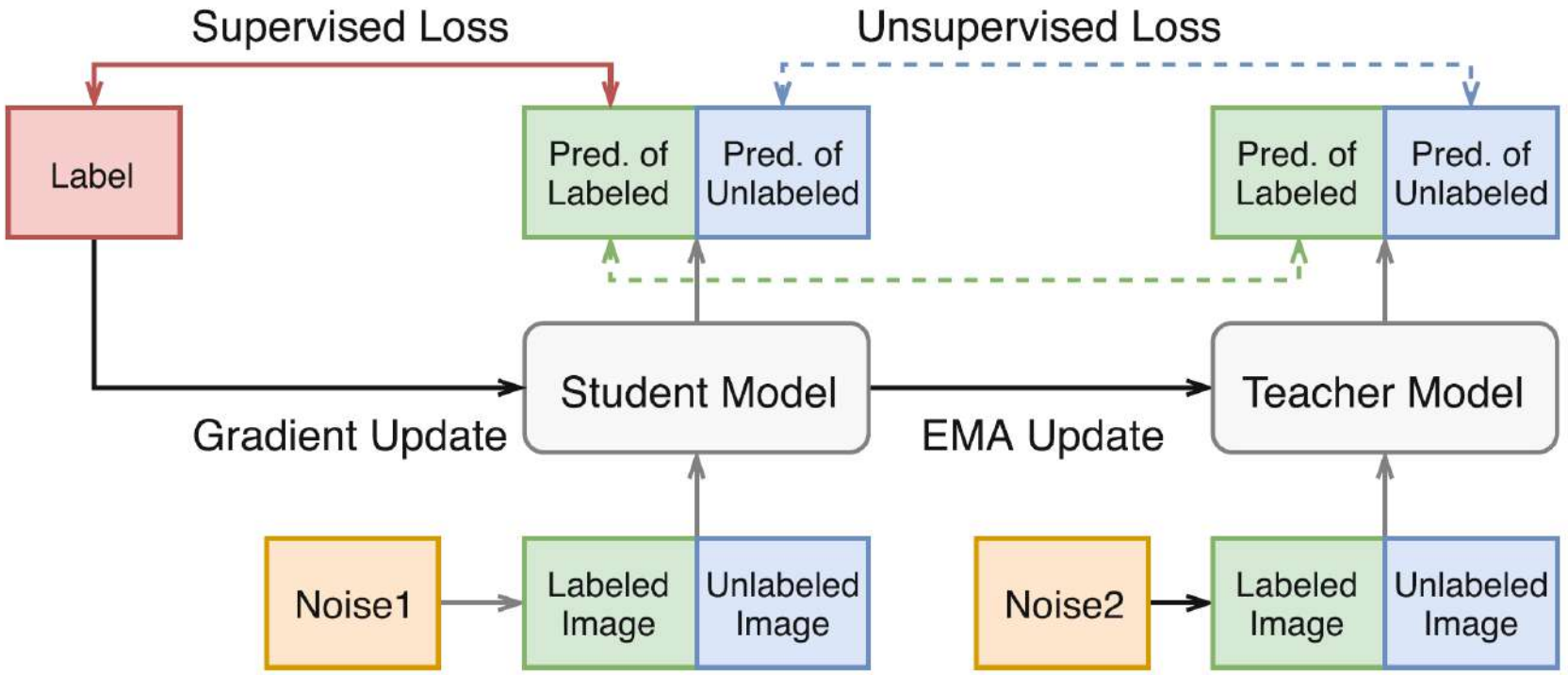
Semi-supervised learning: labelled data + unlabelled data $P(\mathbf{x})$

$P(\mathbf{y} \mid \mathbf{x})$



Laparoscopic video image segmentation



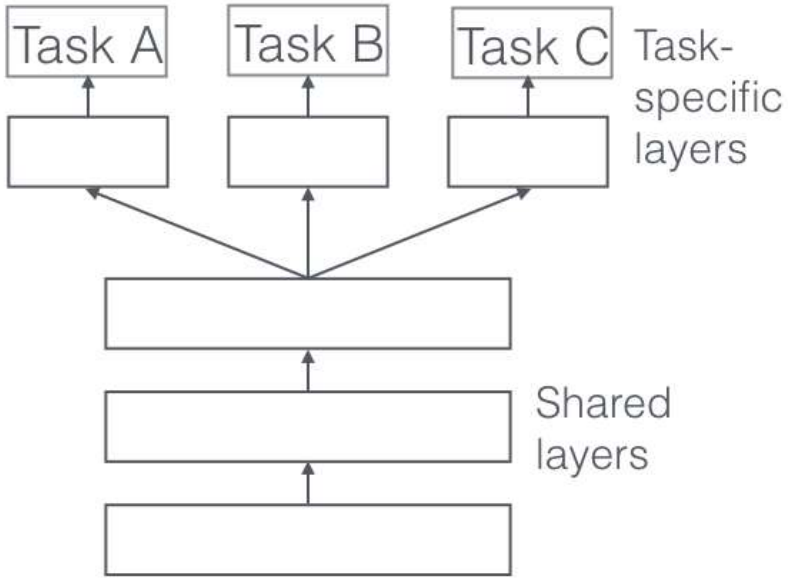


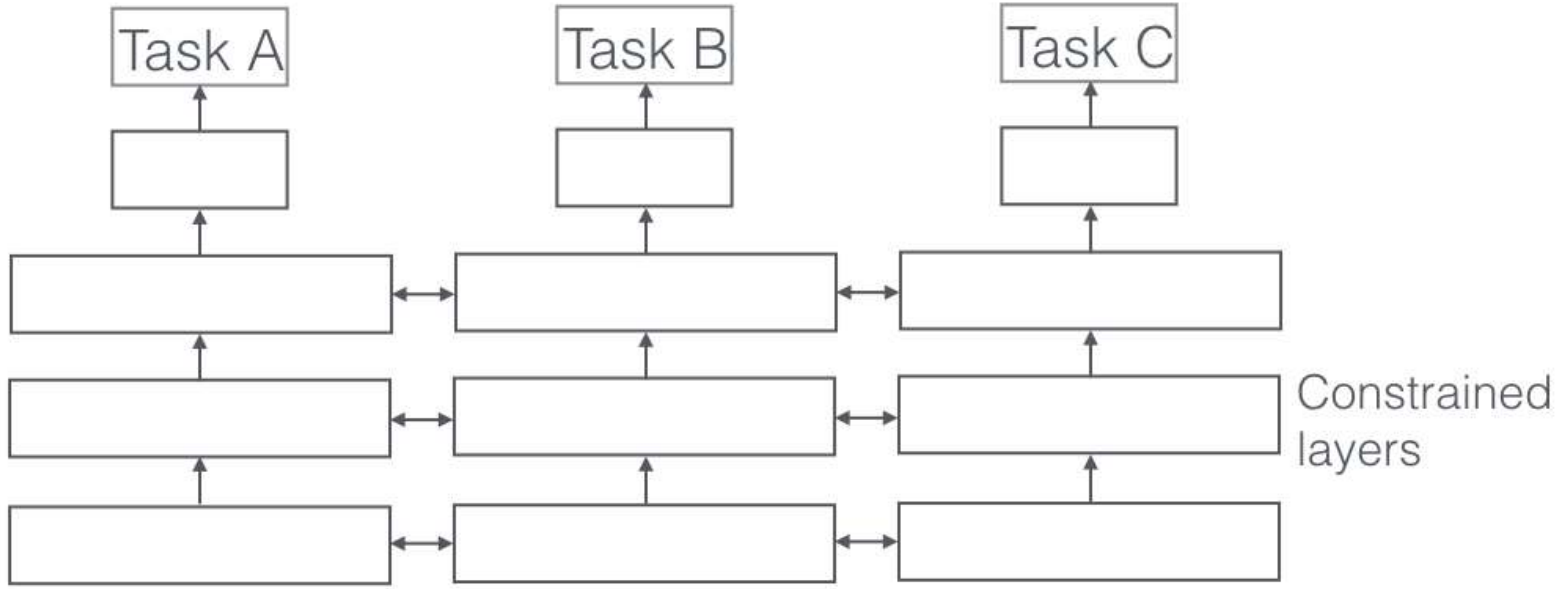
Student model is guided by
- Supervised label
- Teacher's prediction

Teacher model's weight is the moving average of student model.

Random affine transformation is used as noise.

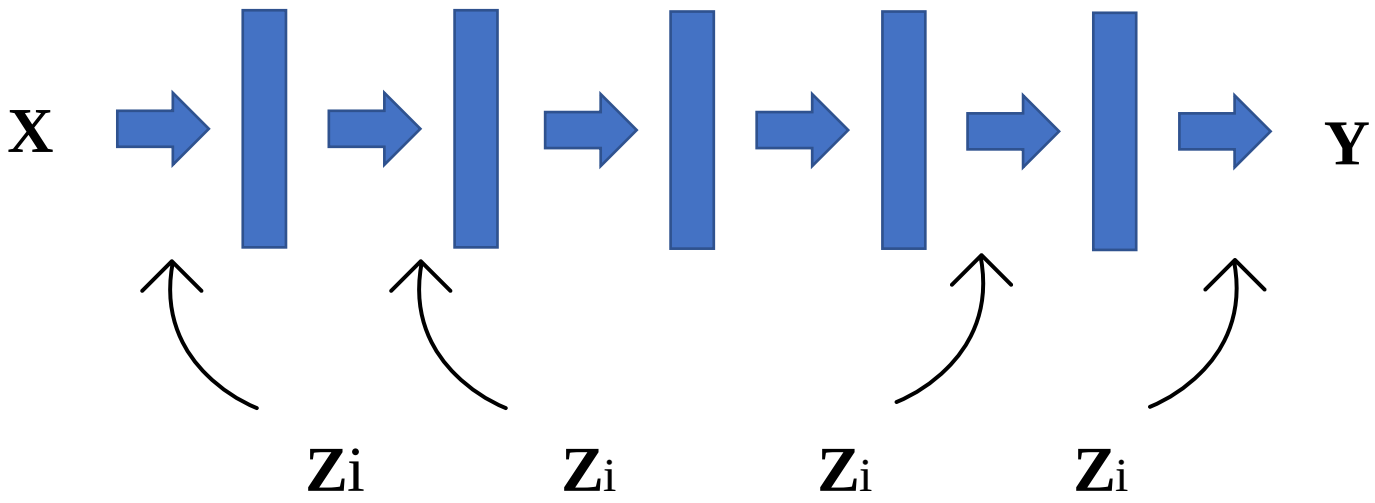
Training and Regularisation| Multi-Task Learning


Task A
Task B
Task C
Task-specific layers
Shared layers


Task A
Task B
Task C
Constrained layers

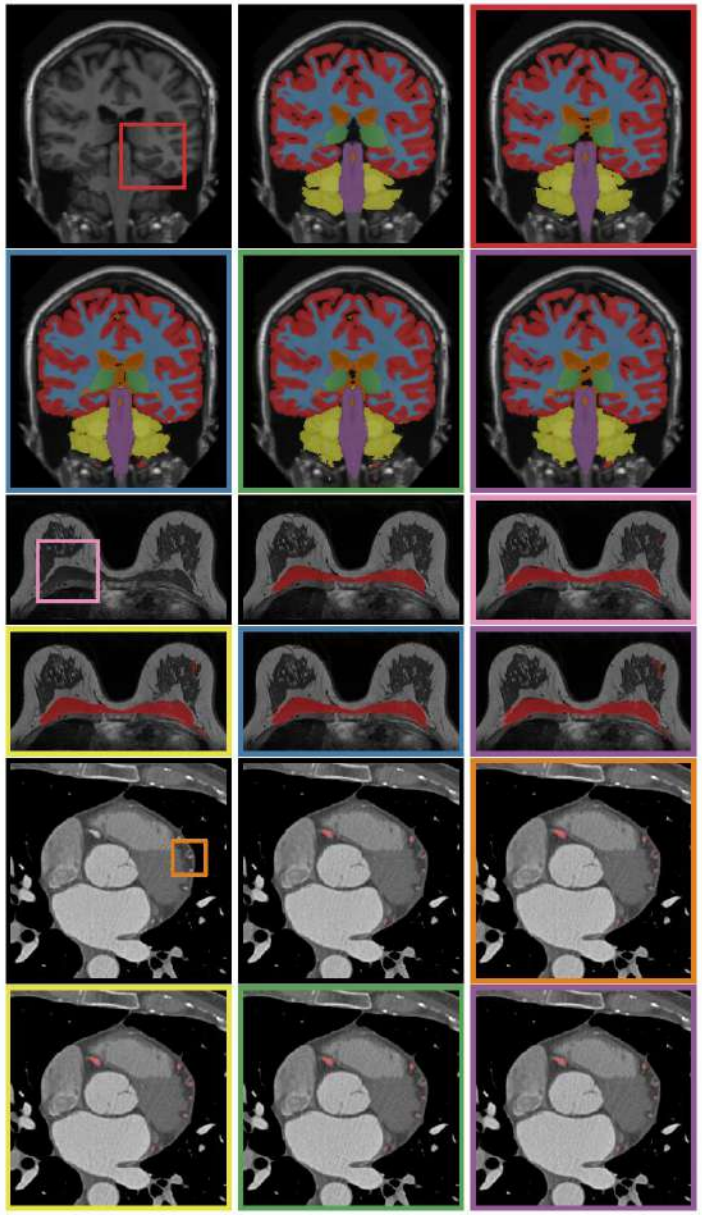
$\mathbf{Z}_i$: One-hot task index / indicator / descriptor

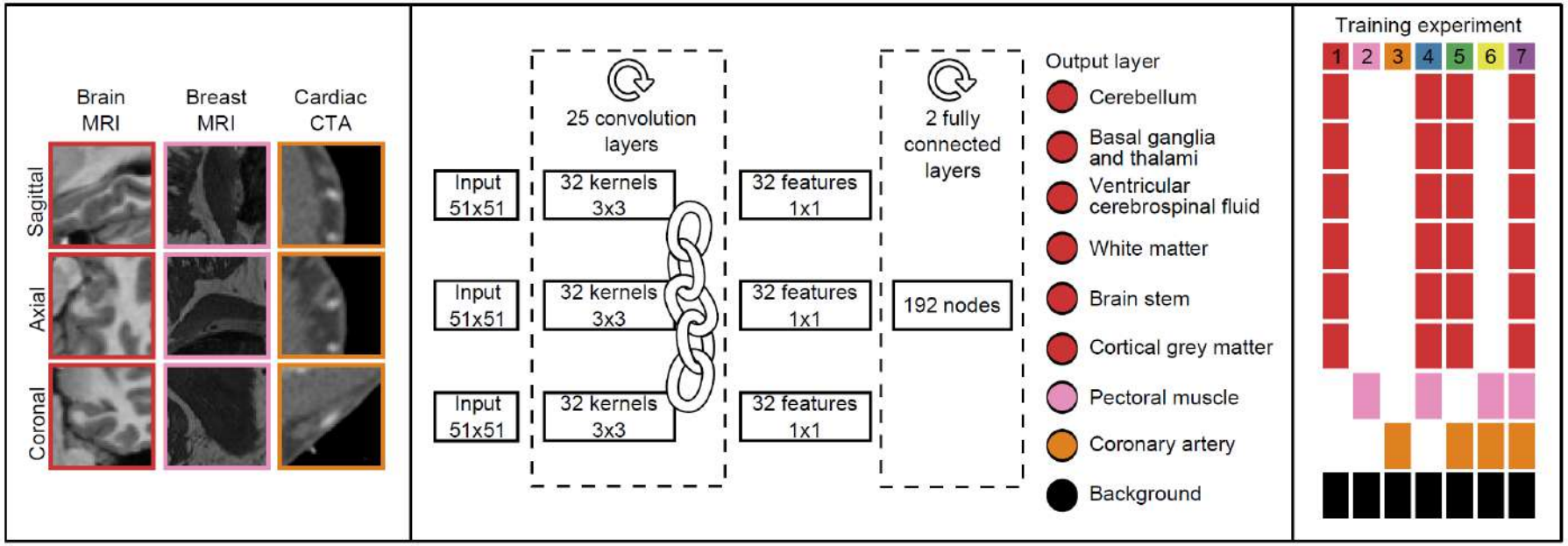
$f(\mathbf{Y}|\mathbf{X}) \Longrightarrow f(\mathbf{Y}|\mathbf{X}, \mathbf{Z}_i)$



- Shared and task specific parameters (negative transfer)
- Concatenation/summation
- Multiplication conditioning (gating/multi-head)

Training and Regularisation| Multi-Task Learning





Links between regularisation methods

- Regularisation, e.g. architecture, optimisation strategies
- Parameter sharing, e.g. semi-supervised, multi-task
- Randomness/noise, e.g. data augmentation, dropout
- Ensemble, e.g. dropout