The University Of Sheffield.

Introduction to Some Parts of Deep Learning

Jiahui Liu
Supervised by Kevin Li Sun

Department of Conputer Science
Faculty of Engineering

Knowledge Distillation (Part 1)

Distilling the knowledge in a neural network

Authors: Hinton G, Vinyals O, Dean J.

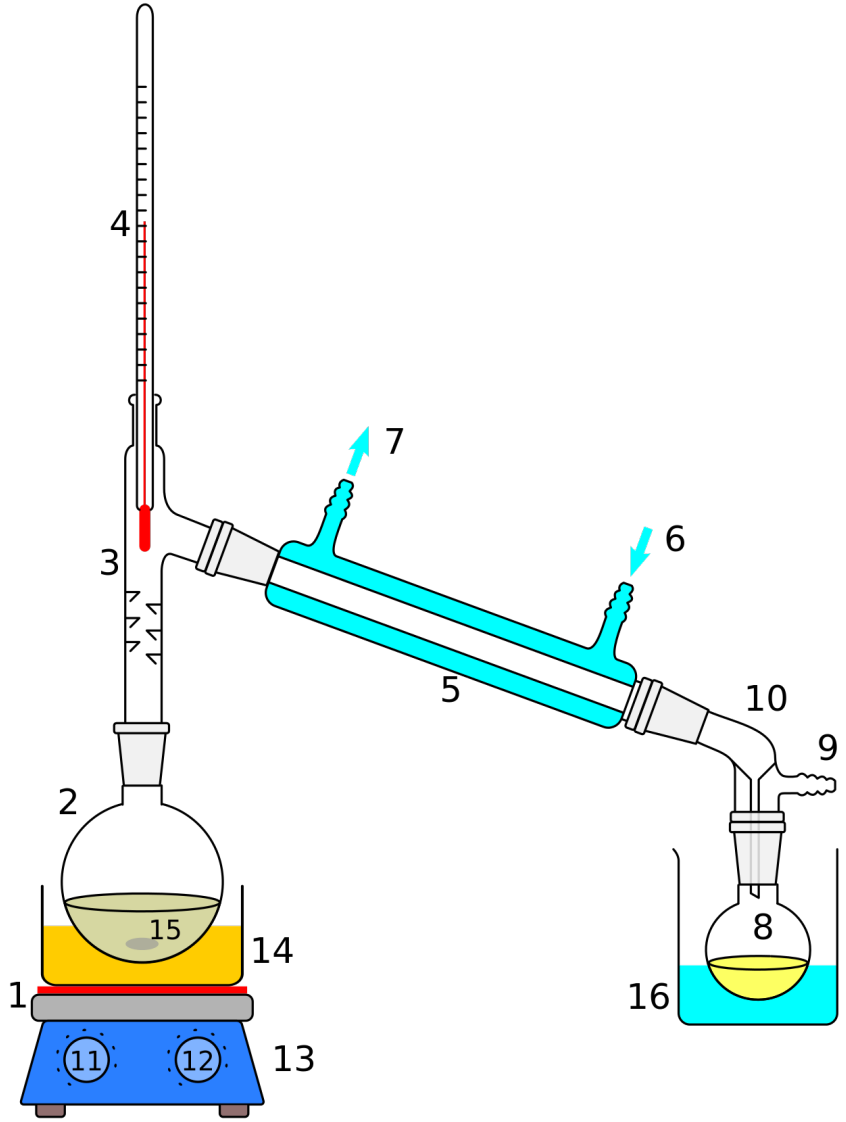
Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015.c

Bayesian Deep Learning (Part 2)

Bayesian Deep Learning

Authors: Yarin Gal

Knowledge Distillation



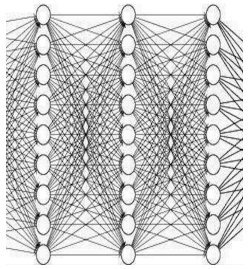
substances

Boiling
(temperature ⇧)

Condensation
(temperature ⇩)

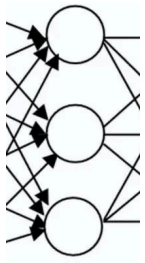
separated components

Knowledge Distillation

a **very complex** neural network ...  ...

Training (T ⇧)

Testing (T ⇩)

a **smaller size** neural network ...  ...

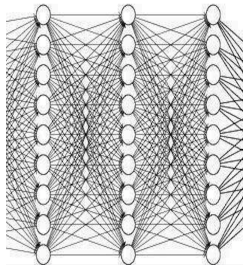
substances

Boiling (temperature ⇧)

Condensation (temperature ⇩)

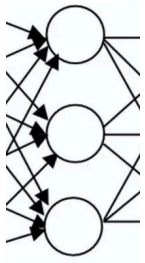
separated components

Knowledge Distillation

a **very complex** neural network ...  ...

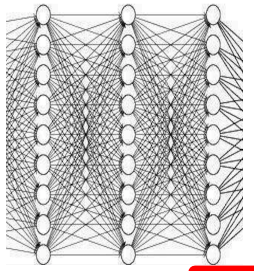
Training ($T \Uparrow$)

Testing ($T \Downarrow$)

a **smaller size** neural network ...  ...

Knowledge Distillation

a **very complex** neural network



Training (T)

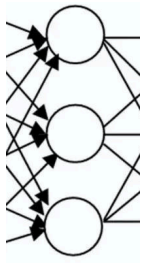
?

Teacher Network (Well trained)

"Knowledge transfer"

Testing (T)

a **smaller size** neural network

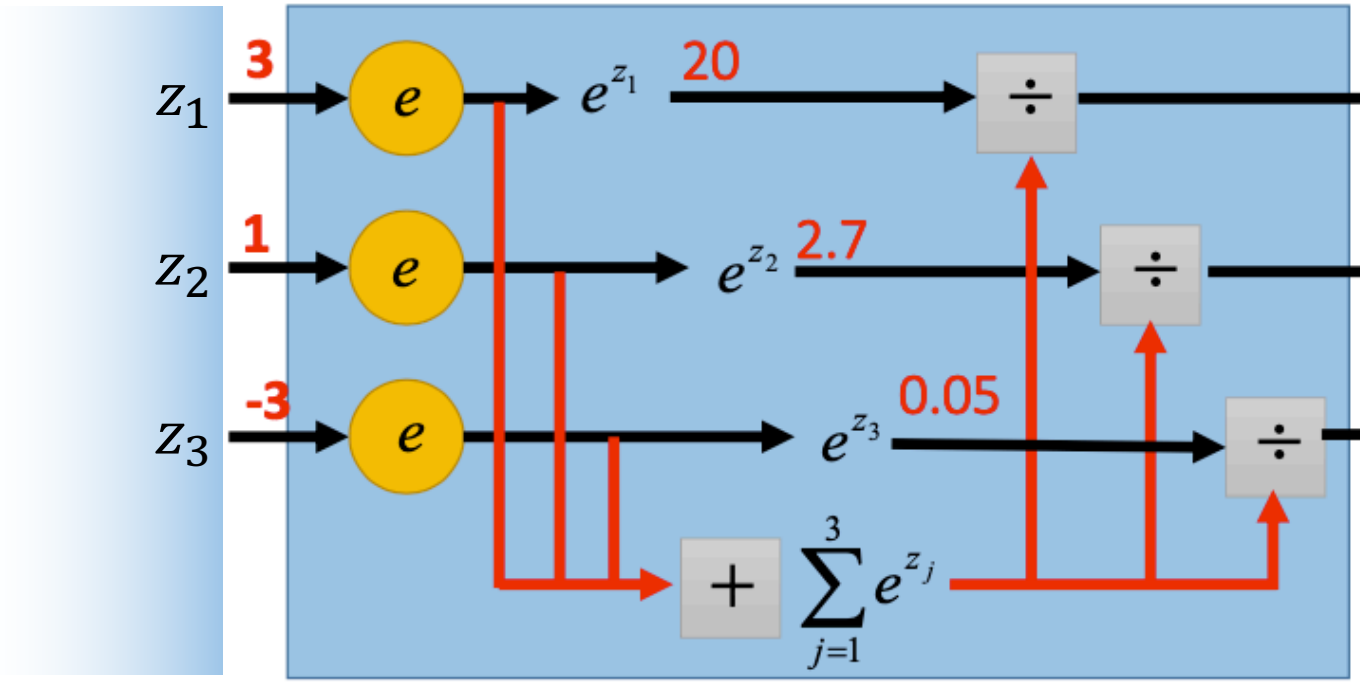


Student Network

Softmax

Softmax Layer

logits



Probability:

- $1 > y_i > 0$
- $\sum_i y_i = 1$

$$y_1 = e^{z_1} \Big/ \sum_{j=1}^{3} e^{z_j} \quad \text{0.88}$$

$$y_2 = e^{z_2} \Big/ \sum_{j=1}^{3} e^{z_j} \quad \text{0.12}$$

$$y_3 = e^{z_3} \Big/ \sum_{j=1}^{3} e^{z_j} \quad \approx 0$$

For example (training):

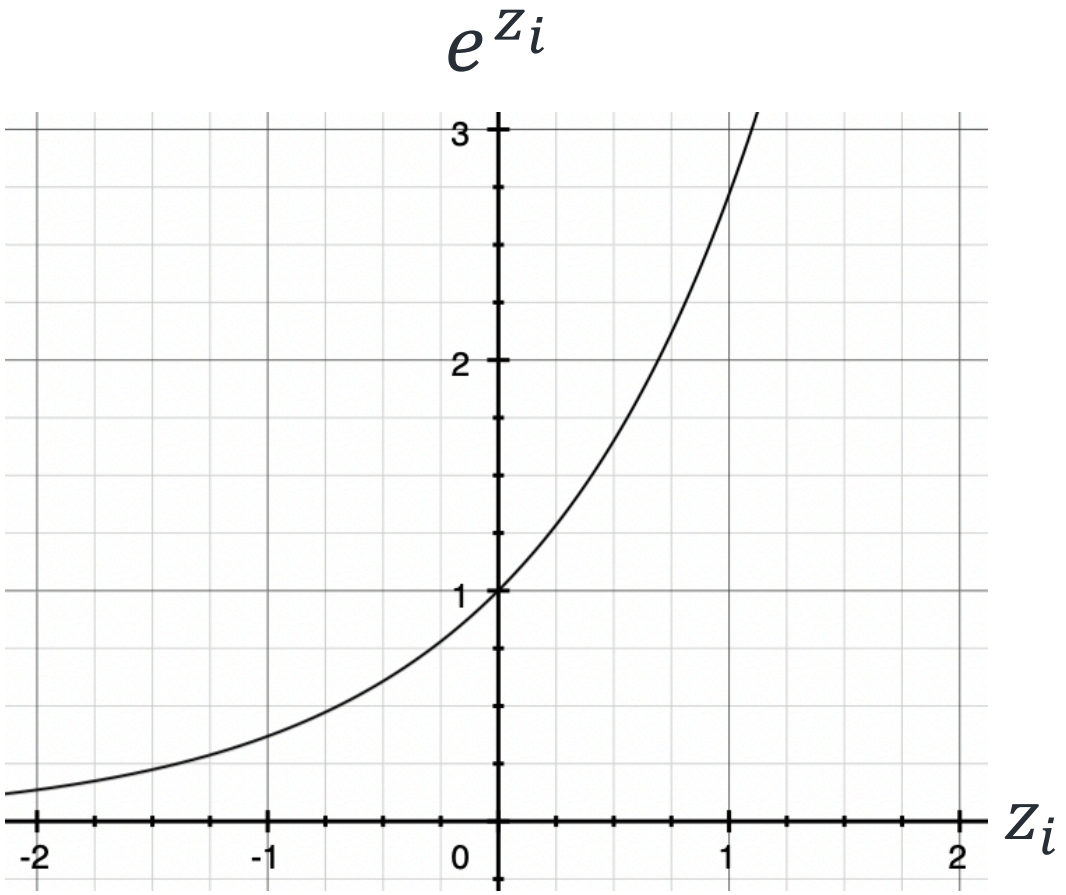
Output	Target
0.88	1
0.12	0
0	0

Loss → **Back Propagation**

Softmax

For the Output:

$$q_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$



"Soft" Output	"Hard" Target
0.88	1
0.12	0
0	0

Loss

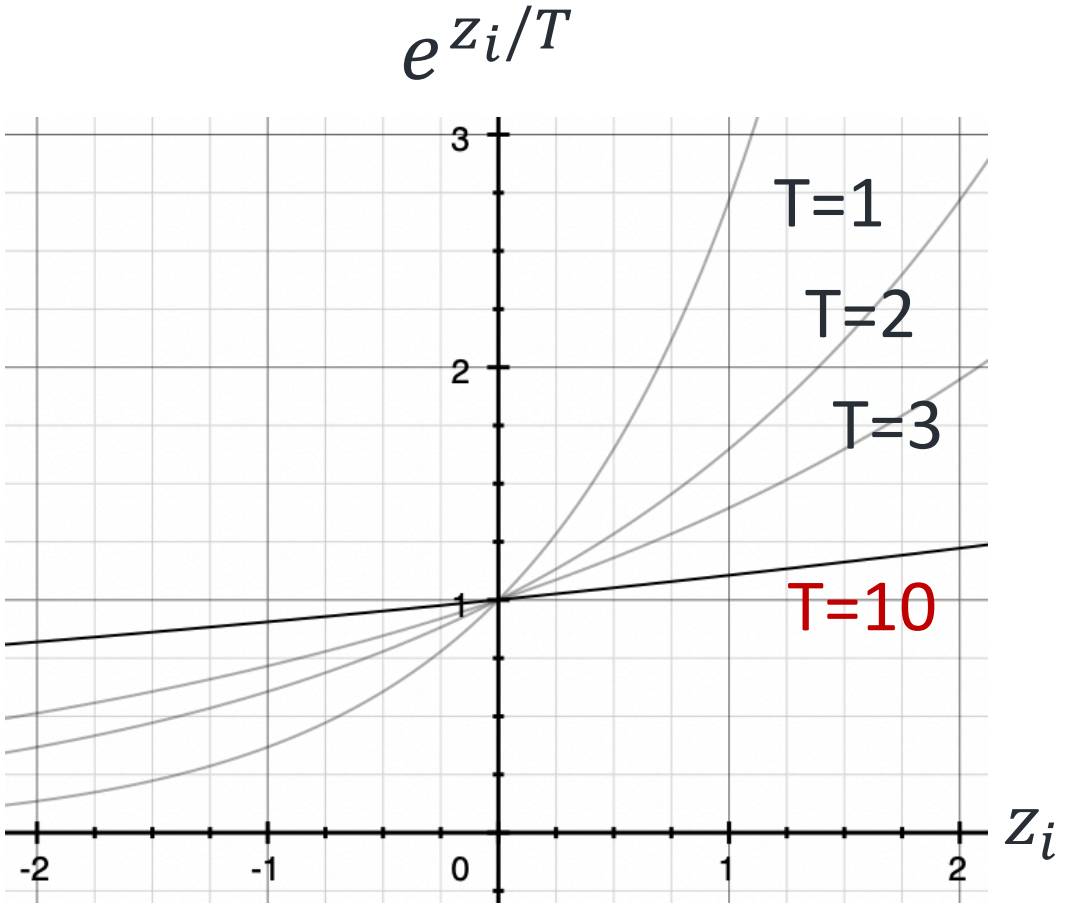
Knowledge Distillation

For the Output:

$$q_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

"Softer"

$$q_i = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$$



"Soft"	"Hard"
Output	Target
0.88	1
0.12	0
0	0

Loss

T -> higher: $e^{z_i/T}$ -> more gentle $\vec{q}$ -> softer

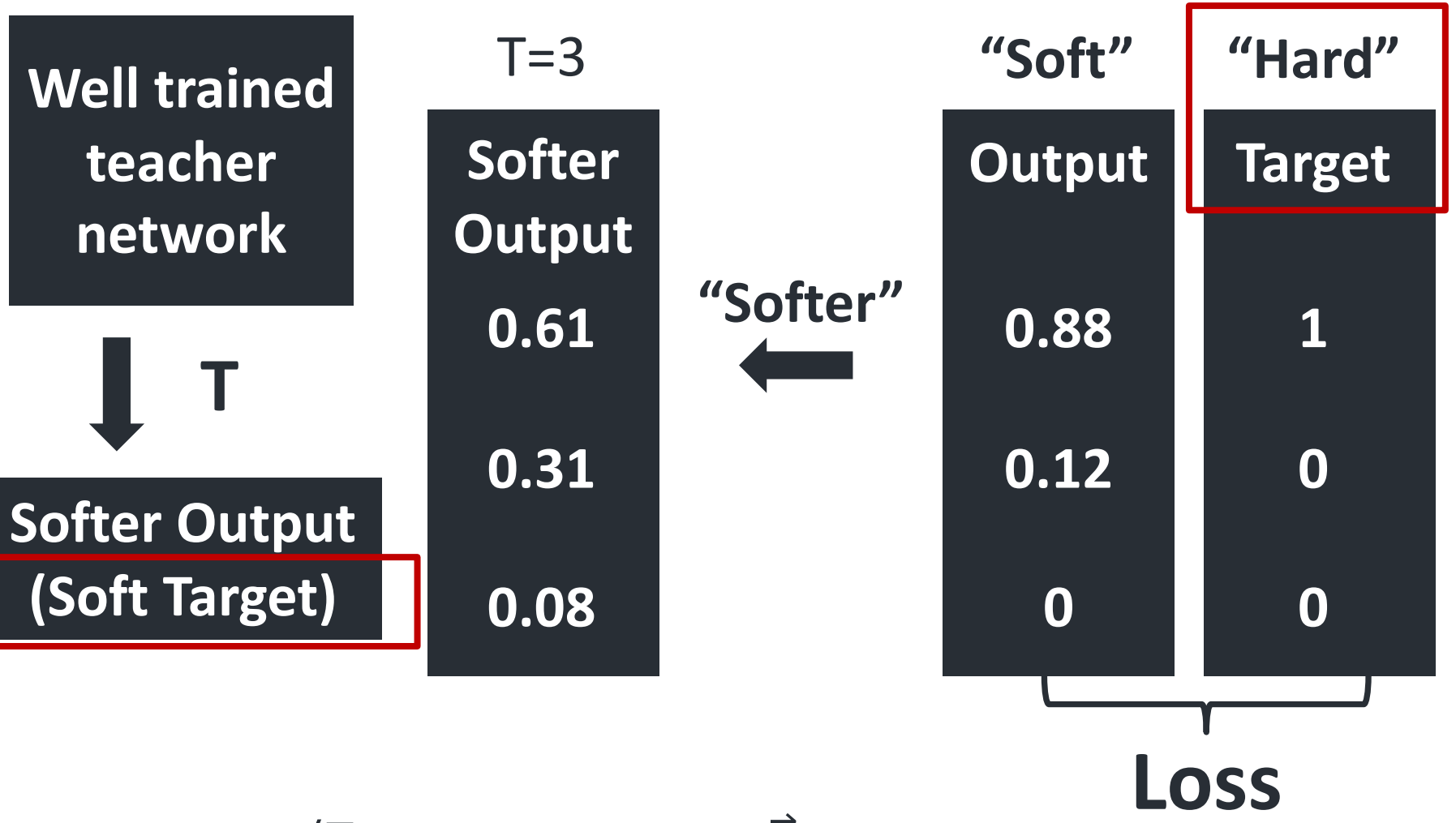
Knowledge Distillation

For the Output:

$$q_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

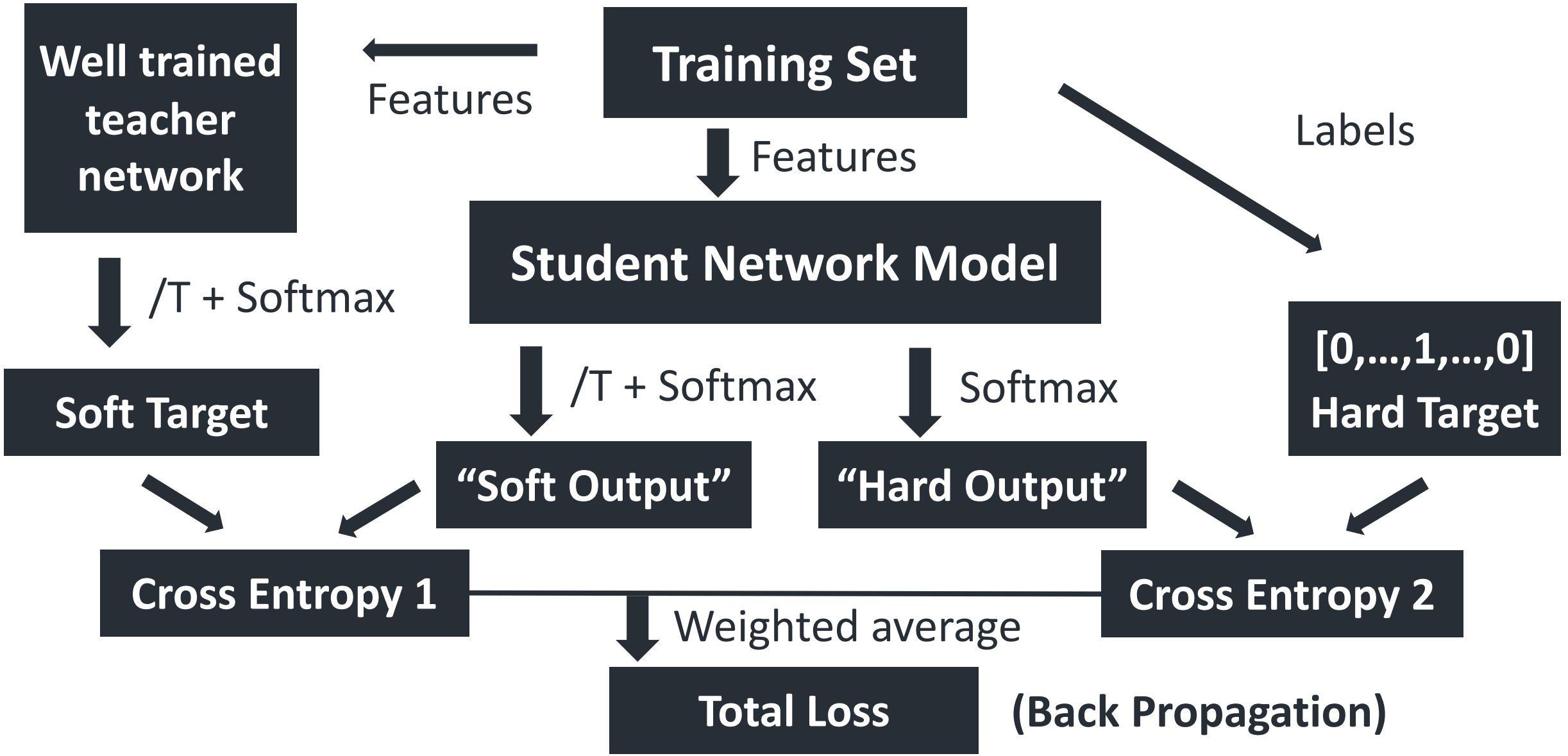
"Softer"

$$q_i = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$$



T -> higher: $e^{z_i/T}$ -> more gentle $\vec{q}$ -> softer

Knowledge Distillation (Train a student network)



Knowledge Distillation

For the Soft Target Loss:

Teacher Network logits: v_i

Student Network logits: z_i

$$p_i = \frac{\exp(v_i/T)}{\sum_j \exp(v_j/T)} \qquad q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

$$C = -\sum_{j=1}^{C} p_j \log q_j$$

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right)$$

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1 + z_i/T}{N + \sum_j z_j/T} - \frac{1 + v_i/T}{N + \sum_j v_j/T}\right)$$

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z_i - v_i)$$

In the high temperature limit, distillation is equivalent to minimizing MSE.

Knowledge Distillation (Part 1)

Distilling the knowledge in a neural network

Authors: Hinton G, Vinyals O, Dean J.

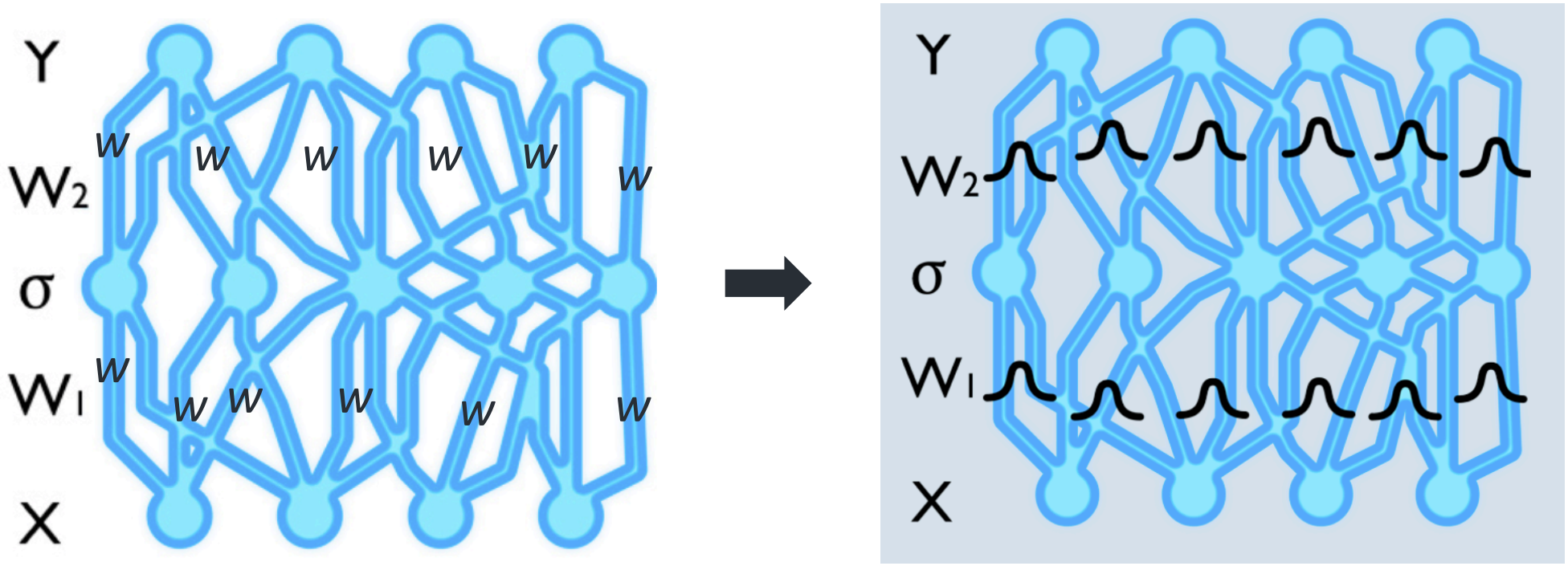
Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015.c

Bayesian Deep Learning (Part 2)

Bayesian Deep Learning

Author: Yarin Gal

Bayesian Deep Learning

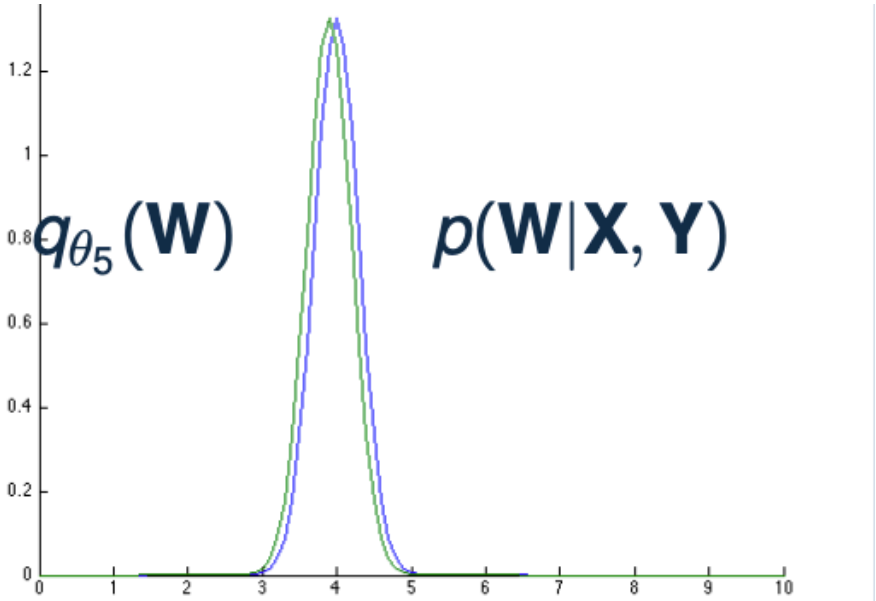


Bayesian Deep Learning

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$$

$$p(w|x, y) = \frac{p(y|x, w)p(w)}{\int p(y|x, w)p(w)dw}$$

Define simple distribution $q_{\mathbf{M}}(\mathbf{W})$
and approximate $q_{\mathbf{M}}(\mathbf{W}) \approx p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$



Variational Inference

KL Divergence (Relative entropy)

Describe the approximation effect of two distributions.

$$p(w|x,y) = \frac{p(y|x,w)p(w)}{\int p(y|x,w)p(w)dw} \qquad q_{\mathbf{M}}(\mathbf{W})$$

$$\begin{aligned} KL(q||p) &= \sum_{i=1}^{n} q(W) log \frac{q(W)}{p(W|X,Y)} = E_q[log \frac{q(W)}{p(W|X,Y)}] \\ &= E_q[logq(W) - logp(W) - logp(Y|X,W) + logp(Y|x)] \\ &= E_q[logq(W) - logp(W)] - E_q[logp(Y|X,W)] + const \end{aligned}$$

Bayesian Deep Learning

Describe the approximation effect of two distributions.

$$p(w|x,y) = \frac{p(y|x,w)p(w)}{\int p(y|x,w)p(w)dw} \qquad q_{\mathbf{M}}(\mathbf{W})$$

$$\begin{aligned} KL(q||p) &= \sum_{i=1}^{n} q(W) log \frac{q(W)}{p(W|X,Y)} = E_q[log \frac{q(W)}{p(W|X,Y)}] \\ &= E_q[logq(W) - logp(W) - logp(Y|X,W) + logp(Y|x)] \\ &= E_q[logq(W) - logp(W)] - E_q[logp(Y|X,W)] + const \end{aligned}$$

$$Loss = E_q[logq(W) - logp(W)] - E_q[logp(Y|X,W)]$$

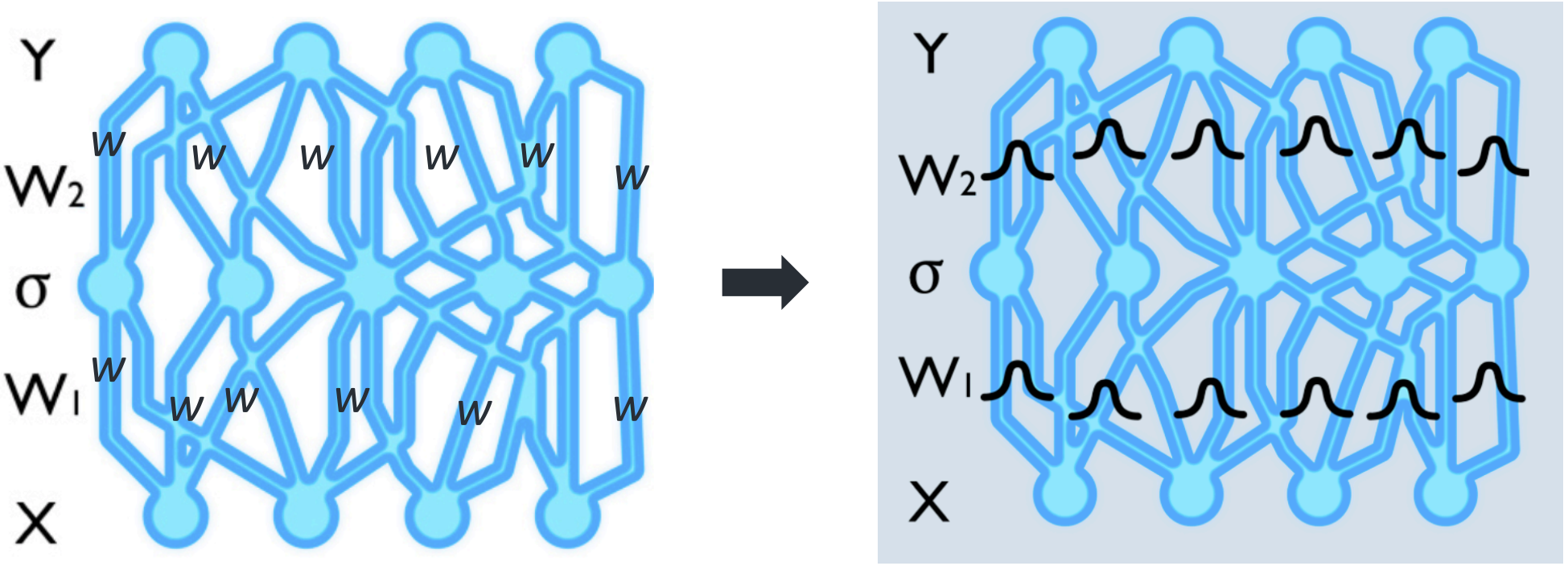
Uncertainty

Aleatoric uncertainty: noise inherent in the data

Epistemic uncertainty: model's lack of knowledge

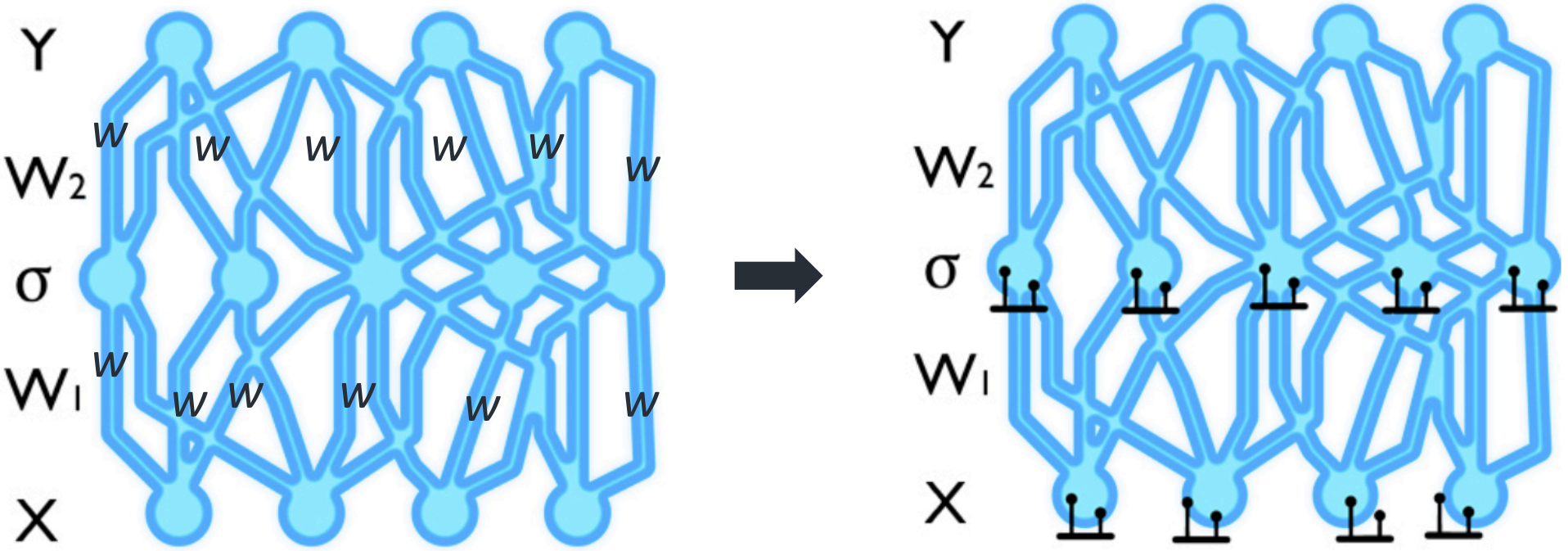
Predictive uncertainty: sum of the two

Bayesian Deep Learning



$q_{\mathbf{M}}(\mathbf{W})$: Gaussian distribution

Bayesian Deep Learning



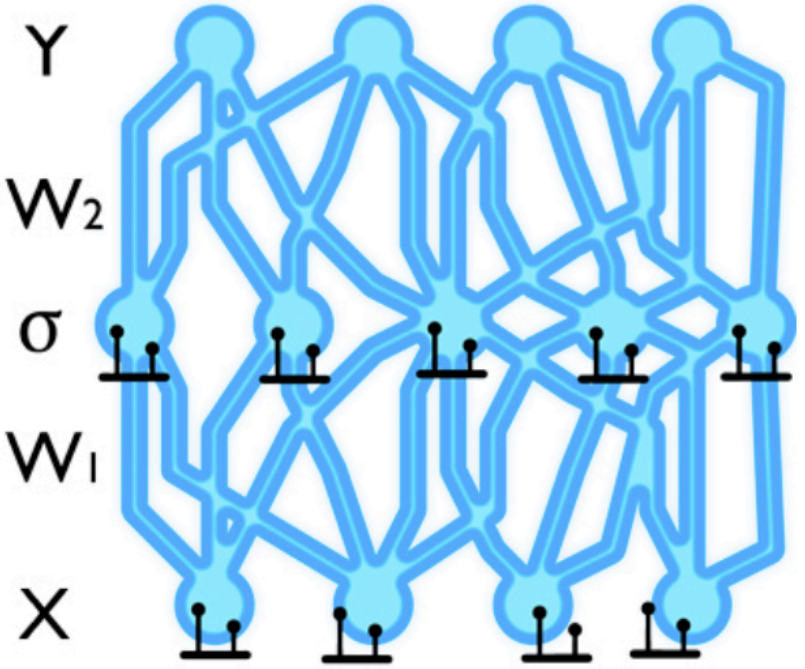
$q_{\mathbf{M}}(\mathbf{W})$: Bernoulli distribution

Bayesian Deep Learning

Variational inference with $\boldsymbol{q_{\mathbf{M}}(\mathbf{W})}$

=

Dropout neural network



$\boldsymbol{q_{\mathbf{M}}(\mathbf{W})}$: Bernoulli distribution

Questions?

The University Of Sheffield.

Thanks for listening

Jiahui Liu
Supervised by Kevin Li Sun
Department of Conputer Science
Faculty of Engineering