Batch normalization

TL;DR:

- It's usually a good idea to normalize linear model inputs

(c) Every machine learning lecturer, ever

Batch normalization

Idea:

- We normalize activation of a hidden layer

  (zero mean unit variance)

$$h_i = \frac{h_i - \mu_i}{\sqrt{\sigma_i^2}}$$

- Update μ_i, σ_i^2 with moving average while training

$$\mu_i := \alpha \cdot mean_{batch} + (1-\alpha) \cdot \mu_i$$

$$\sigma_i^2 := \alpha \cdot variance_{batch} + (1-\alpha) \cdot \sigma_i^2$$

Batch normalization

Idea:

– We normalize activation of a hidden layer

(zero mean unit variance)

$$h_i = \frac{h_i - \mu_i}{\sqrt{\sigma_i^2}}$$

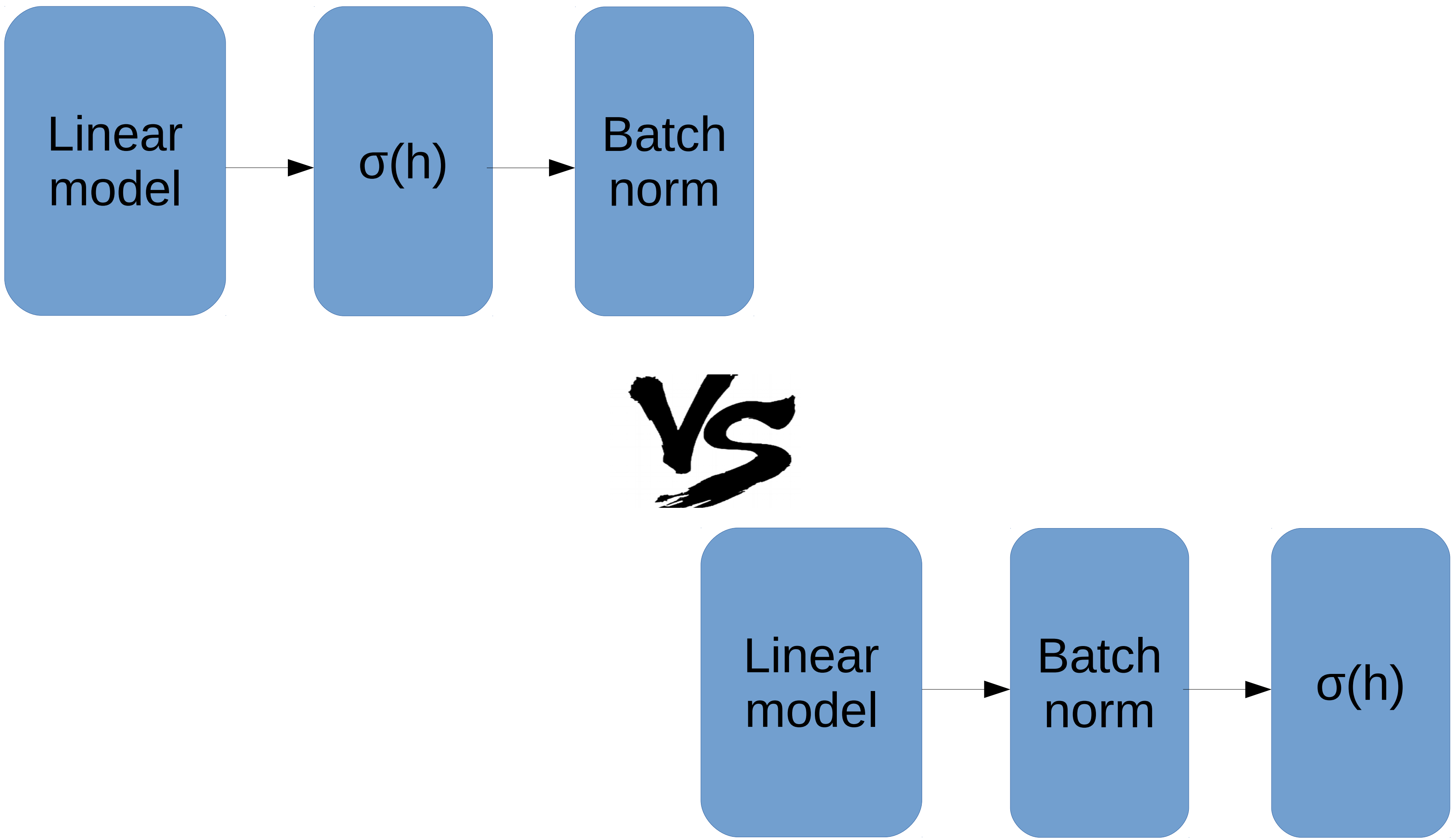
i stands for i-th neuron

– Update μ_i, σ_i^2 with moving average while training

$$\mu_i := \alpha \cdot mean_{batch} + (1-\alpha) \cdot \mu_i$$

$$\sigma_i^2 := \alpha \cdot variance_{batch} + (1-\alpha) \cdot \sigma_i^2$$

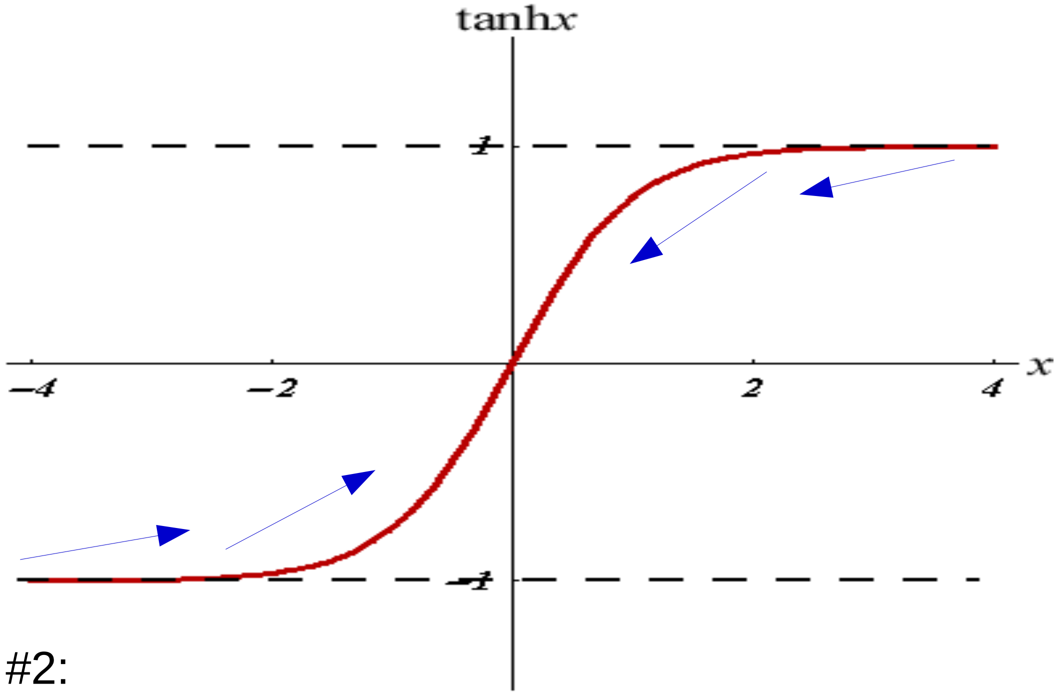
Batch normalization



Batch normalization

Good side effect #1:

- Vanishing gradient less a problem for sigmoid-like nonlinearities



Good side effect #2:

- We no longer need to train bias (+b term in Wx+b)

Weight normalization

Same problem, different solution

- Learn separate "direction" w and "length" l

$$\hat{w} \stackrel{\text{def}}{=} \frac{w}{||w||} \cdot l$$

- Much simpler, but requires good init

More normalization

Layer/Instance normalization

- Like batchnorm, but normalizes over different axes

Normprop

- A special training algorithm

Self-normalizing neural networks (SELU)

Nuff

Coding time!