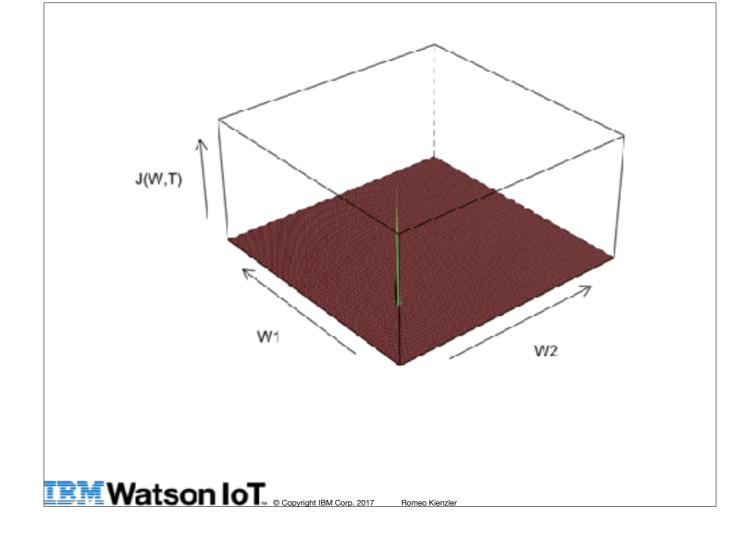
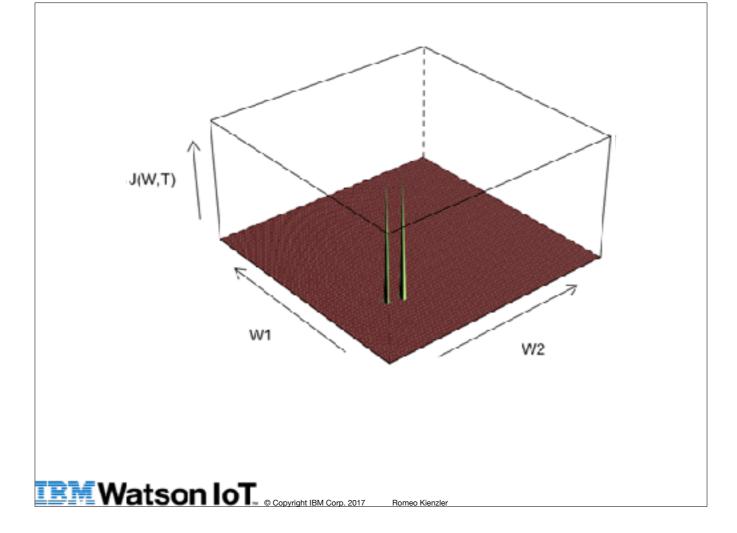
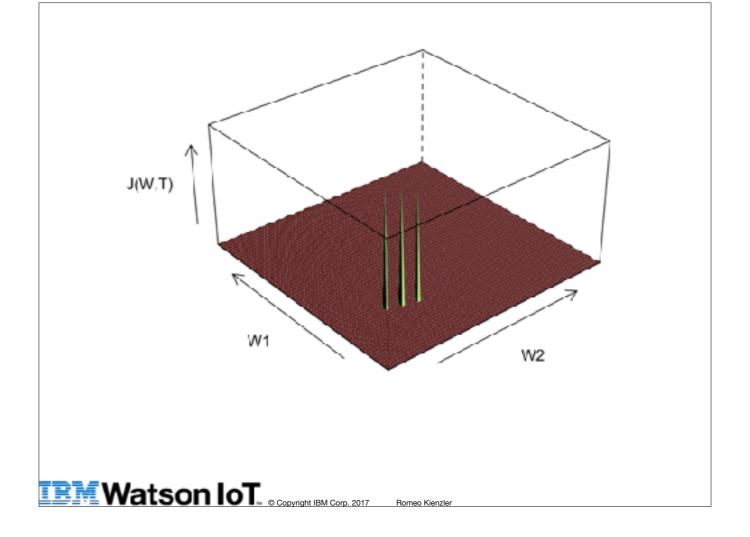
Gradient Descent Updater Strategies

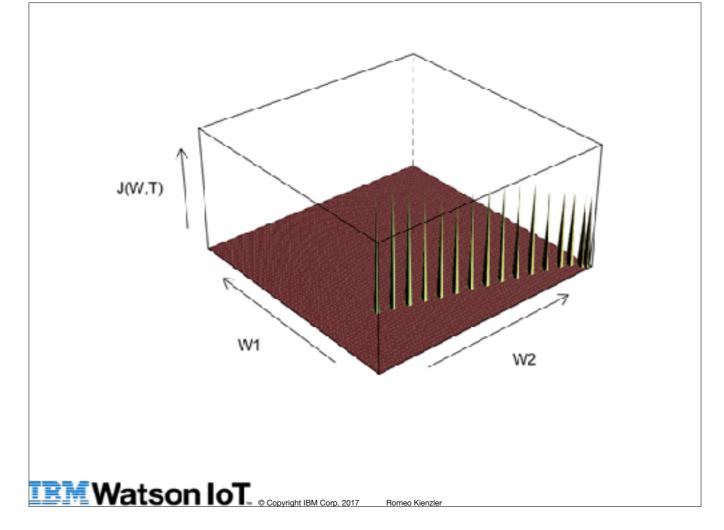




We then could climb down the ladder



..one step by another...



..until we reached an optimal value - this method is called gradient descent - and with all it's variations it's the de-facto standard in neural network training.

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t, X, Y)$$

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t, X, Y)$$

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t, X, Y)$$

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t, X, Y)$$

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t(X, Y))$$

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t, X, Y)$$

Watson IoT © Copyright IBM Corp. 2017

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t, X, Y)$$

Stochastic Gradient Descent

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t, x^{(i)}, y^{(i)})$$

Stochastic Gradient Descent

$$\theta_{t+1} = \theta_t - \eta \Delta_\theta J(\theta_t, x^{(i)}, y^{(i)})$$

(Mini) Batch Gradient Descent

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t, x^{(i..i+n)}, y^{(i..i+n)})$$

(Mini) Batch Gradient Descent

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta} J(\theta_t(x^{(i..i+n)}, y^{(i..i+n)}))$$

Momentum

$$\begin{aligned} \nu_t &= \gamma \nu_{t-1} - \eta \Delta_\theta J(\theta_t, X, Y) \\ \theta_{t+1} &= \theta_t - \nu_t \end{aligned}$$

Momentum

$$\nu_t = \gamma \nu_{t-1} - \eta \Delta_{\theta} J(\theta_t, X, Y)$$

$$\theta_{t+1} = \theta_t - \nu_t$$

Watson IoT © Copyright IBM Corp. 2017

Domoo Kionzlor

Momentum

$$\nu_t = \gamma \nu_{t-1} - \eta \Delta_{\theta} J(\theta_t, X, Y)$$

$$\theta_{t+1} = \theta_t (-\nu_t)$$

Watson IoT © Copyright IBM Corp. 2017

Romeo Kienzler

Nesterov accelerated gradient

$$\nu_t = \gamma \nu_{t-1} - \eta \Delta_{\theta} J(\theta_t - \gamma \nu_{t-1}, X, Y)$$

$$\theta = \theta - \nu_t$$

Nesterov accelerated gradient

$$\nu_t = \gamma \nu_{t-1} - \eta \Delta_{\theta} J(\theta_t - \gamma \nu_{t-1}) X, Y)$$

$$\theta = \theta - \nu_t$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \Delta_{\theta} J(\theta_{t,i}, X, Y)$$

Watson IoT © Copyright IBM Corp. 2017 Rom

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \Delta_{\theta} J(\theta_{t,i}, X, Y)$$

Watson IoT. © Copyright IBM Corp. 2017

Romeo Kienzle

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \Delta_{\theta} J(\theta_{t,i}, X, Y)$$

Watson IoT. © Copyright IBM Corp. 2017

Romeo Kienzler

$$\theta_{t+1,i} = \theta_{t,i} - \underbrace{\sqrt{G_{t,ii} + \epsilon}}_{\eta} \Delta_{\theta} J(\theta_{t,i}, X, Y)$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \epsilon}} \Delta_{\theta} J(\theta_{t,i}, X, Y)$$

Watson IoT © Copyright IBM Corp. 2017

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G(i) + \epsilon}} \Delta_{\theta} J(\theta_{t,i}, X, Y)$$

$$\theta_{t+1} = \theta_{t} - \frac{\eta}{\sqrt{G_{t} + \epsilon}} \Delta_{\theta} J(\theta_{t}, X, Y)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \Delta_{\theta} J(\theta_t, X, Y)$$

Watson IoT @ Copyright IBM Corp. 2017

Adadelta

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[\Delta\theta^2]_t + \epsilon}} \Delta_{\theta} J(\theta_t, X, Y)$$

Adadelta

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[\Delta\theta^2]_t + \epsilon}} \Delta_\theta J(\theta_t, X, Y)$$

RMSProp

RMSProp Adam

AdaMax **RMSProp**

Adam



AdaMax

RMSProp Adam

Nadam



Summary

Activation Functions