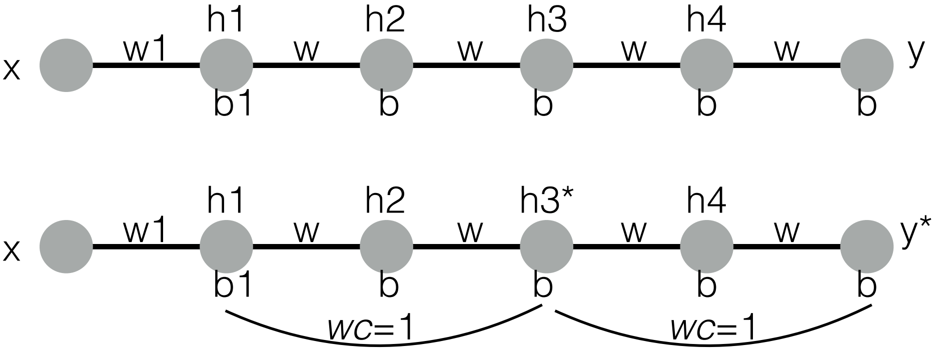
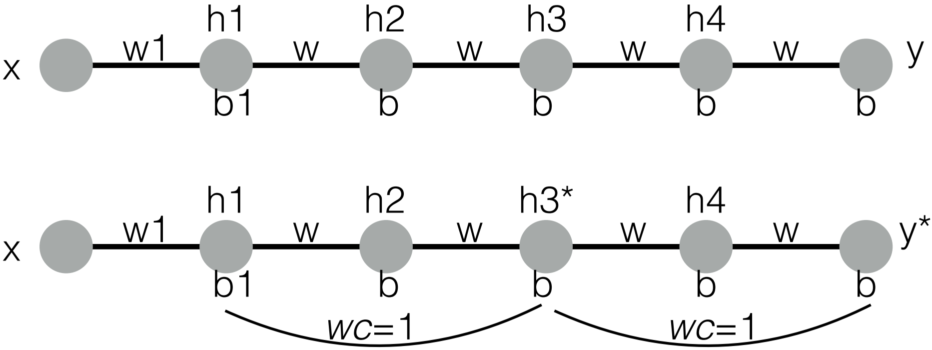
Assignment 2

Due week 6 before class time



Two networks shown above with identical weights, the top network is a linear chain and the second network is the same except for two additional short circuit connections. Short circuit connections have weights fixed to *wc*=1

Show that |dy/dw1| <= |dy*/dw1| &
|dy/db1| <= |dy*/db1|



The connections are defined such that

h1 = \sigma(w1 x + b1)

h2 = \sigma(w h1 + b)

h3 = \sigma(w h2 + b)

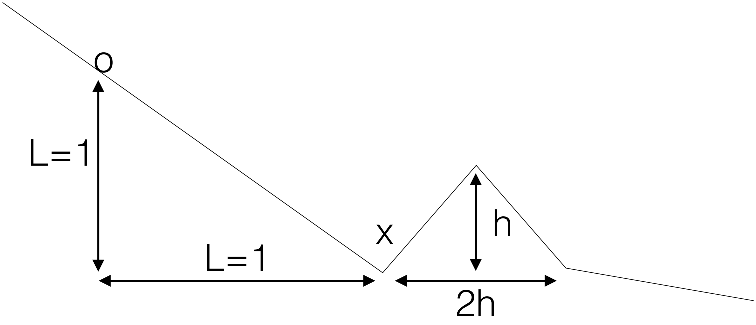
h4 = \sigma(w h3 + b) or \sigma(w h3*+b)

y = \sigma(w h4 + b)

h3* = \sigma(w h2 + h1 + b)

y* = \sigma(w h4 + h3* + b)

where activation function is such that \sigma' > 0



The diagram above shows a plot of a 1D function and gradient descend is applied to minimise the function at the point ‘o’. there is a bump a distance L away with bump dimensions given as h x 2h. Let L = 1, a = 0.3 and h > a where a is the learning rate

(1) what will happen if you apply standard gradient descend? (Ans: stuck at point ‘x’)

(2) if you apply adam optimisation with parameters given in the next slide, what is the max height ‘h’ of the bump in which the adam optimiser will escape the local min at ‘x’? use \epsilon = 0 instead of \epsilon = 1e-8 in your calculations.

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

You are given the code

comp_tree.py

Which generates this tree given on the right. Label all the nodes