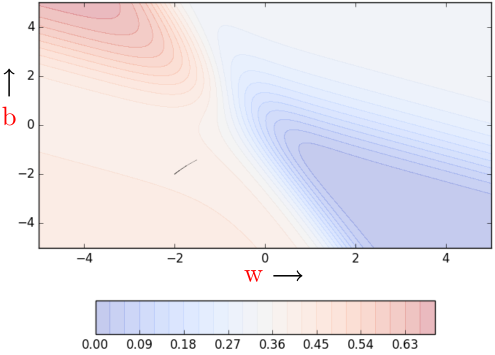
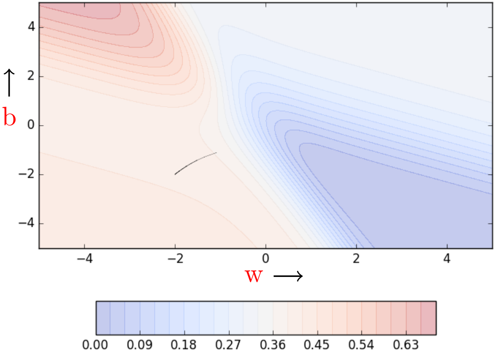
Module 5.7 : Tips for Adjusting learning Rate and Momentum

Before moving on to advanced optimization algorithms let us revisit the problem of learning rate in gradient descent

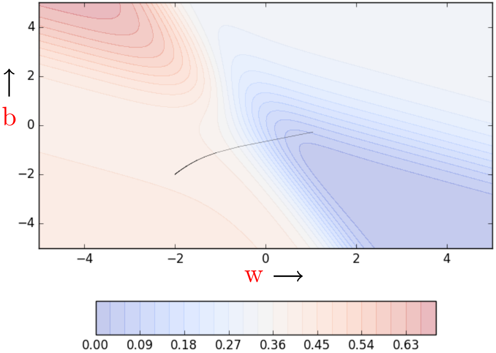
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)



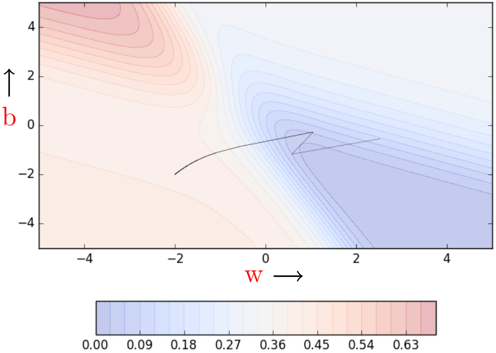
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



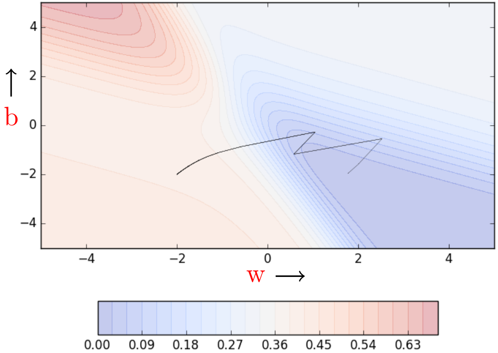
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



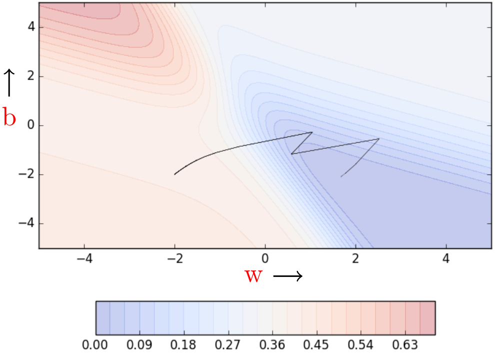
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



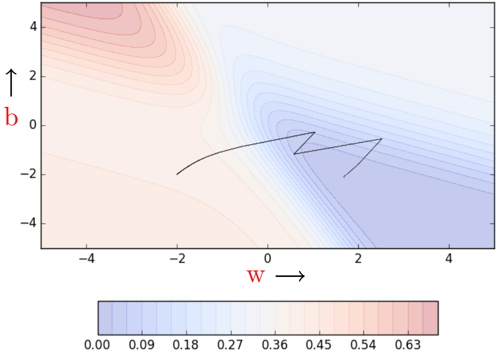
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



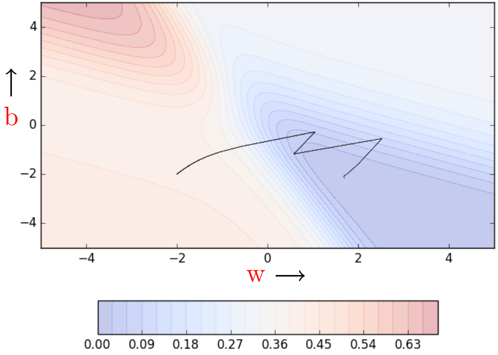
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



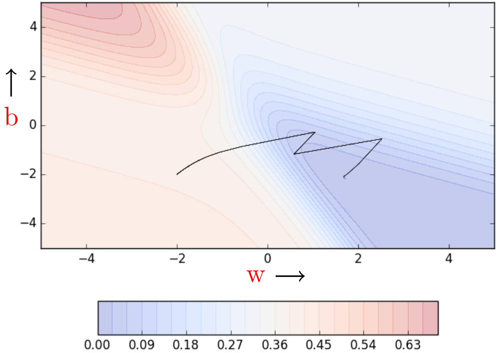
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



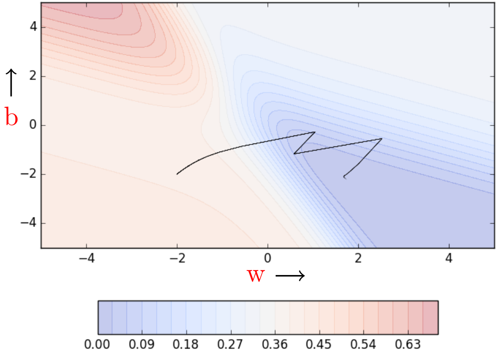
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



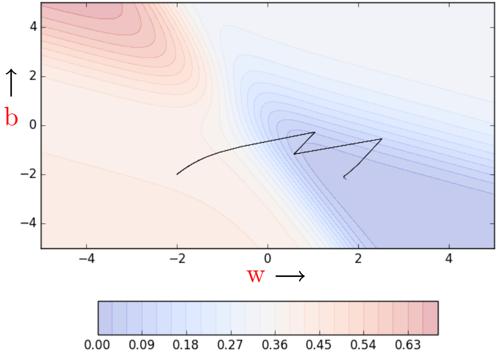
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



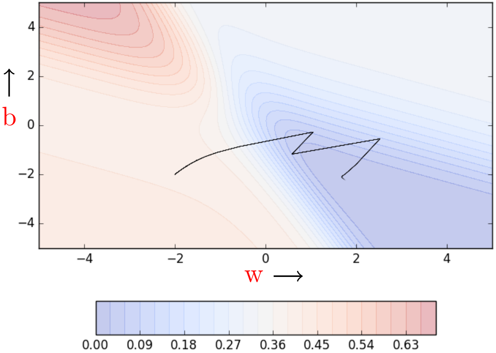
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



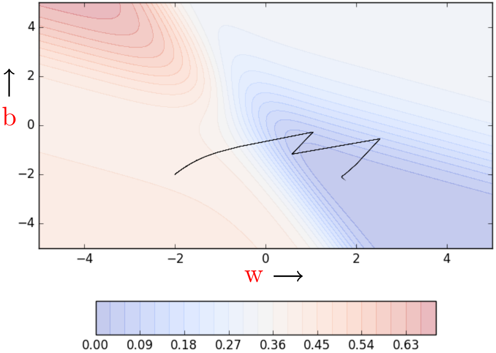
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



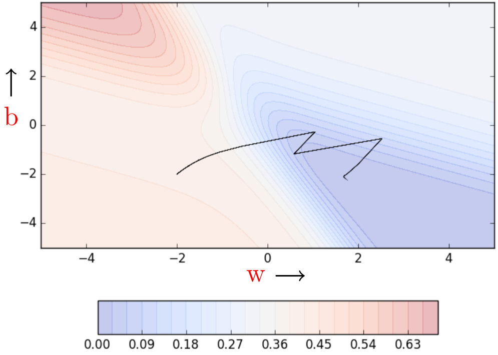
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



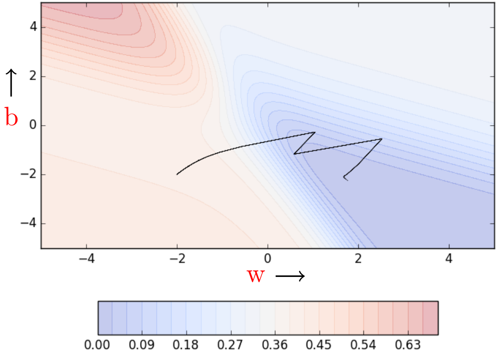
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



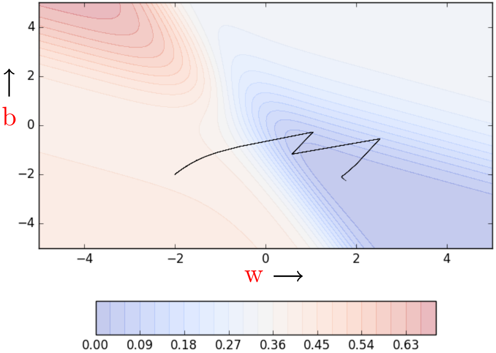
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



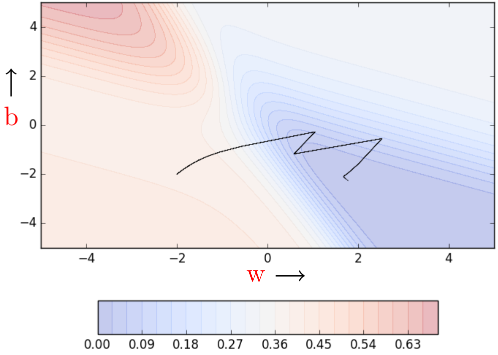
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



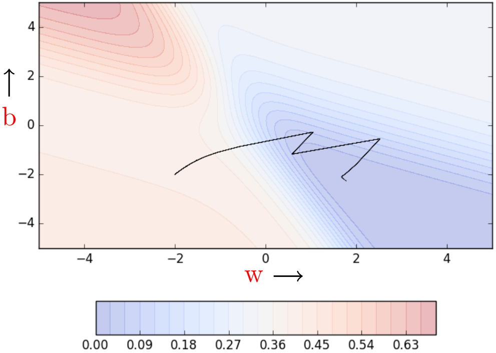
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



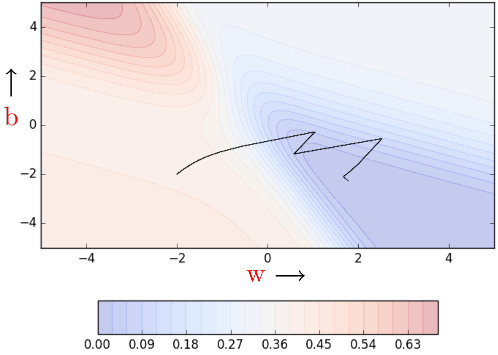
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10



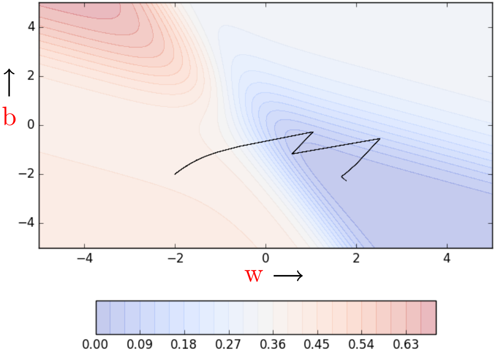
- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10
- On the regions which have a steep slope, the already large gradient blows up further



- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10
- On the regions which have a steep slope, the already large gradient blows up further
- It would be good to have a learning rate which could adjust to the gradient ...



- One could argue that we could have solved the problem of navigating gentle slopes by setting the learning rate high (i.e., blow up the small gradient by multiplying it with a large η)
- Let us see what happens if we set the learning rate to 10
- On the regions which have a steep slope, the already large gradient blows up further
- It would be good to have a learning rate which could adjust to the gradient ... we will see a few such algorithms soon



Tips for initial learning rate ?

Tips for initial learning rate ?

- Tune learning rate [Try different values on a log scale: 0.0001, 0.001, 0.01, 0.1. 1.0]

Tips for initial learning rate ?

- Tune learning rate [Try different values on a log scale: 0.0001, 0.001, 0.01, 0.1. 1.0]
- Run a few epochs with each of these and figure out a learning rate which works best

Tips for initial learning rate ?

- Tune learning rate [Try different values on a log scale: 0.0001, 0.001, 0.01, 0.1. 1.0]
- Run a few epochs with each of these and figure out a learning rate which works best
- Now do a finer search around this value [for example, if the best learning rate was 0.1 then now try some values around it: 0.05, 0.2, 0.3]

Tips for initial learning rate ?

- Tune learning rate [Try different values on a log scale: 0.0001, 0.001, 0.01, 0.1. 1.0]
- Run a few epochs with each of these and figure out a learning rate which works best
- Now do a finer search around this value [for example, if the best learning rate was 0.1 then now try some values around it: 0.05, 0.2, 0.3]
- Disclaimer: these are just heuristics ... no clear winner strategy

Tips for annealing learning rate

Tips for annealing learning rate

- **Step Decay:**

Tips for annealing learning rate

- **Step Decay:**
    - Halve the learning rate after every 5 epochs or

Tips for annealing learning rate

- **Step Decay:**
  - Halve the learning rate after every 5 epochs or
  - Halve the learning rate after an epoch if the validation error is more than what it was at the end of the previous epoch

Tips for annealing learning rate

- **Step Decay:**
  - Halve the learning rate after every 5 epochs or
  - Halve the learning rate after an epoch if the validation error is more than what it was at the end of the previous epoch
- **Exponential Decay:** $\eta = \eta_0^{-kt}$ where η_0 and k are hyperparameters and t is the step number

Tips for annealing learning rate

- **Step Decay:**
  - Halve the learning rate after every 5 epochs or
  - Halve the learning rate after an epoch if the validation error is more than what it was at the end of the previous epoch
- **Exponential Decay:** $\eta = \eta_0^{-kt}$ where η_0 and k are hyperparameters and t is the step number
- **1/t Decay:** $\eta = \frac{\eta_0}{1+kt}$ where η_0 and k are hyperparameters and t is the step number

Tips for momentum

- The following schedule was suggested by Sutskever *et. al.*, 2013

$$\mu_t = min(1 - 2^{-1-log_2(\lfloor t/250 \rfloor + 1)}, \mu_{max})$$

where, μ_{max} was chosen from $\{0.999, 0.995, 0.99, 0.9, 0\}$