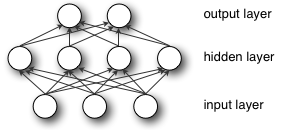
Random Initialization in Neural Network

Breaking symmetry is essential here, and not for the reason of performance. Imagine first 2 layers of multilayer perceptron (input and hidden layers):



During forward propagation each unit in hidden layer gets signal:

enter image description here

That is, each hidden unit gets sum of inputs multiplied by the corresponding weight.

Now imagine that you initialize all weights to the same value (e.g. zero or one). In this case, **each hidden unit will get exactly the same signal**. E.g. if all weights are initialized to 1, each unit gets signal equal to sum of inputs (and outputs sigmoid(sum(inputs))). If all weights are zeros, which is even worse, every hidden unit will get zero signal. **No matter what was the input - if all weights are the same, all units in hidden layer will be the same too**.

This is the main issue with symmetry and reason why you should initialize weights randomly (or, at least, with different values). Note, that this issue affects all architectures that use each-to-each connections.

**Analogy:**

I hope it's a good analogy. I've tried to explain it as simple as possible.

Imagine that someone has dropped you from a helicopter to an unknown mountain top and you're trapped there. Everywhere is fogged. The only thing you know is that you should get down to the sea level somehow. Which direction should you take to get down to the lowest possible point?

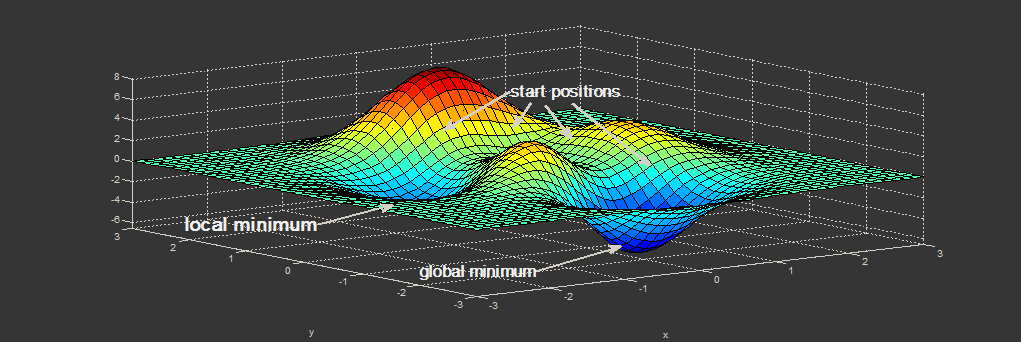
If you couldn't find a way to the sea level and so the helicopter would take you again and would drop you to the same mountain top position. You would have to take the same directions again because you're "initializing" yourself to the same **starting positions**.

However, each time the helicopter drops you somewhere **random** on the mountain, you would take different directions and steps. So, there would be a **better chance** for you to reach to the lowest possible point.

This is what is meant by **breaking the symmetry**. The initialization is **asymmetric** (which is different) so you can find different solutions to the same problem.

In this analogy, **where you land is the weights**. So, with different weights, there's a better chance of reaching to the lowest (or lower) point.

Also, it increases the **entropy** in the system so the system can create more information to help you find the lower points (local or global minimums).

[](https://i.stack.imgur.com/EGx2a.png)