

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

[Take the 2-minute tour](#)

## java simple neural network setup

**CAREERS 2.0**  
by stackoverflow



Have projects on Codeplex?  
Import them easily to your profile

I have decided to play around with some simple concepts involving neural networks in Java, and in adapting somewhat useless code I found on a forum, I have been able to create a very simple model for the typical beginner's XOR simulation:

```
public class MainApp {
    public static void main (String [] args) {
        Neuron xor = new Neuron(0.5f);
        Neuron left = new Neuron(1.5f);
        Neuron right = new Neuron(0.5f);
        left.setWeight(-1.0f);
        right.setWeight(1.0f);
        xor.connect(left, right);

        for (String val : args) {
            Neuron op = new Neuron(0.0f);
            op.setWeight(Boolean.parseBoolean(val));
            left.connect(op);
            right.connect(op);
        }

        xor.fire();

        System.out.println("Result: " + xor.isFired());
    }
}
```

```

public class Neuron {
    private ArrayList inputs;
    private float weight;
    private float threshold;
    private boolean fired;

    public Neuron (float t) {
        threshold = t;
        fired = false;
        inputs = new ArrayList();
    }

    public void connect (Neuron ... ns) {
        for (Neuron n : ns) inputs.add(n);
    }

    public void setWeight (float newWeight) {
        weight = newWeight;
    }

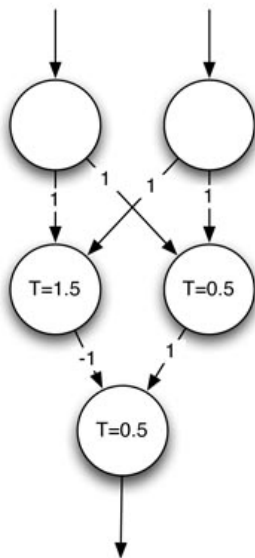
    public void setWeight (boolean newWeight) {
        weight = newWeight ? 1.0f : 0.0f;
    }

    public float getWeight () {
        return weight;
    }

    public float fire () {
        if (inputs.size() > 0) {
            float totalWeight = 0.0f;
            for (Neuron n : inputs) {
                n.fire();
                totalWeight += (n.isFired()) ? n.getWeight() : 0.0f;
            }
        }
    }
}

```

In my main class, I've created the simple simulation in modeling Jeff Heaton's diagram:



However, I wanted to ensure my implementation for the Neuron class is correct..I've already tested all possible inputs ( [true true], [true false], [false true], [false false]), and they all passed my manual verification. Additionally, since this program accepts the inputs as arguments, it also seems to pass manual verification for inputs such as [true false false], [true true false], etc..

But conceptually speaking, would this implementation be correct? Or how can I improve upon it before I start further development and research into this topic?

Thank you!

[java](#) [artificial-intelligence](#) [neural-network](#) [simulation](#)

asked Jan 18 '11 at 1:18



jerluc

2,494 9 26

[add comment](#)

## 2 Answers

It looks like a good starting point. I do have a few suggestions:

1. For scalability, `fire()` should be restructured so that a neuron that's already fired with the current input set doesn't have to recalculate each time. This would be the case if you had another hidden layer, or more than one output node.
2. Consider splitting your threshold calc into its own method. Then you can subclass `Neuron` and use different types of activation functions (bipolar sigmoid, RBF, linear, etc).
3. To learn more complex functions, add a bias input to each neuron. It's basically like another input with its own weight value, but the input is always fixed at 1 (or -1).
4. Don't forget to allow for training methods. Backpropagation will need something like the inverse of `fire()`, to take a target output and ripple the weight changes through each layer.

answered Jan 18 '11 at 3:45



[darkcanuck](#)

221 1 2

---

Thanks for your suggestions! However, I will need to do a lot more researching on the subject, being that I feel far from knowing what most of what you suggested means haha – [jerluc](#) Jan 18 '11 at 4:30

---

[add comment](#)

**CAREERS 2.0**  
by stackoverflow



>



Easily apply for your dream job  
No formatting needed!

From the (limited) work I've done with neural nets, that implementation and model looks correct to me - the output is what I'd expect and the source looks solid.

answered Jan 18 '11 at 1:59



[berry120](#)

21.3k 4 52 106

---

Thanks for your response, from any experience you have, are there any problems you see in the scalability of this `Neuron` class? From my perspective, I've tried to make it scalable enough to accommodate for multiple inner layers of neurons, but two perspectives are better than one. – [jerluc](#) Jan 18 '11 at 2:38

---

[add comment](#)

---

Not the answer you're looking for? Browse other questions tagged [java](#)

[artificial-intelligence](#) [neural-network](#) [simulation](#) or [ask your own question](#).