The Gentle Way To Start With Quantum Machine Learning

Quantum Bayesian Networks are probably what you're looking for

Frank Zickert | Quantum Machine Learning 20 hours ago · 4 min read ★

Do you want to get started with Quantum Machine Learning? Have a look at **Hands-On Quantum Machine Learning With Python**.

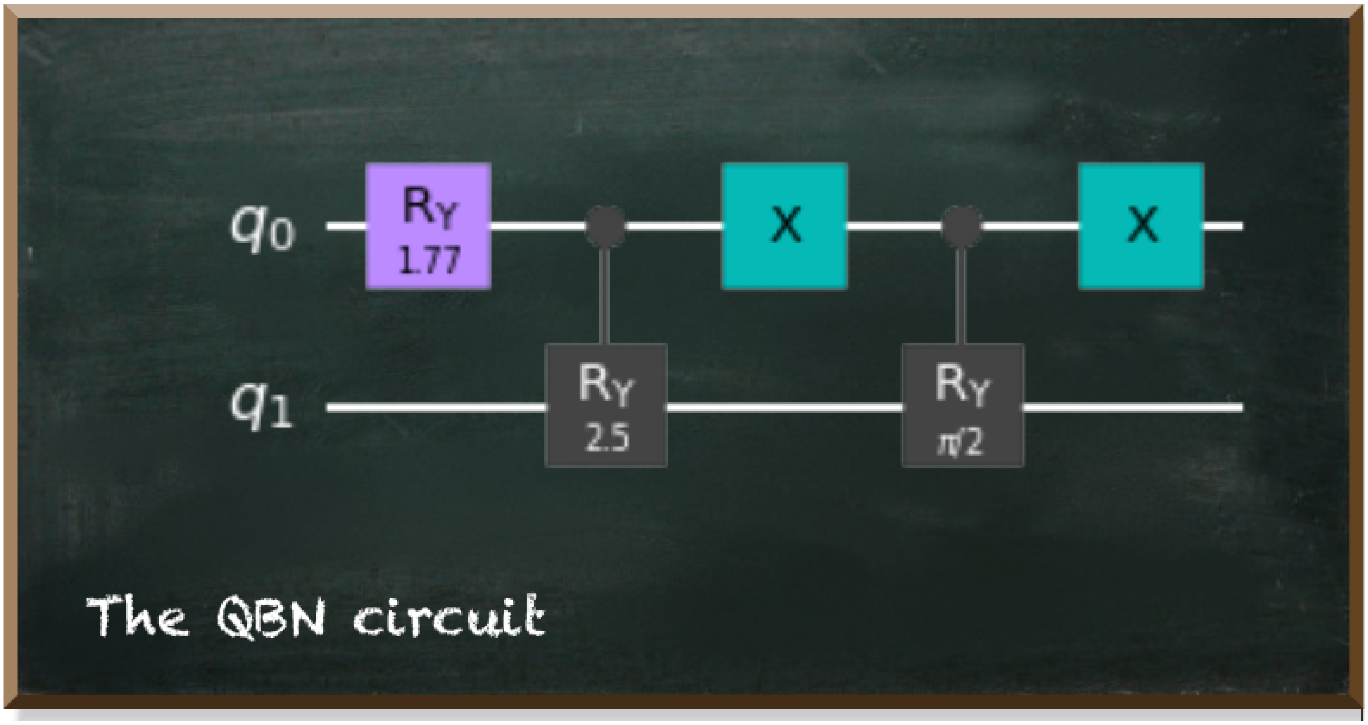


Image by author

If you want to get started with quantum machine learning, you'll love quantum Bayesian networks (QBN). They are intuitive, and thus, easy to understand. Yet, they use the fundamental quantum computing concepts. Therefore, you can learn the concepts hands-on.

In my weekly quantum machine learning challenge, I asked a simple question: *"Can a student solve the problem?"*

It turns out that the solution is a quantum Bayesian network.

"What?!" you're thinking, *"this challenge barely contains any information at all. Not to speak of some handles to decide on how to solve it, and finally to solve it!"*

Of course, you're right. I provided more information when I formulated the challenge.

But, this post is not about the challenge itself. If you're interested in the challenge, you can find it here. Instead, this post is about how a simple quantum Bayesian network teaches you the fundamental concepts of quantum computing and machine learning.

When I ask whether the student can solve the problem, you might be tempted to say yes or no. Either answer could be correct. And that's the point. We don't know unless the student tried to solve it.

Until then, we can only speculate. To answer the question, we need to express our beliefs. So, at best, our answer is probabilistic. For example, it could be *"I believe the student can solve the problem"* or *"she has a chance of 50% to solve it."*

Bayesian networks and quantum computing share the probabilistic perspective.

The simplest variable we can think of is Boolean. That is a variable with two possible values. It is true or false, 1 or 0, or the student solves the problem or not. In a Bayesian network, we use a Bernoulli distribution to describe a Boolean variable. It denotes the probabilities of either value. The following figure shows an exemplary Bernoulli distribution.

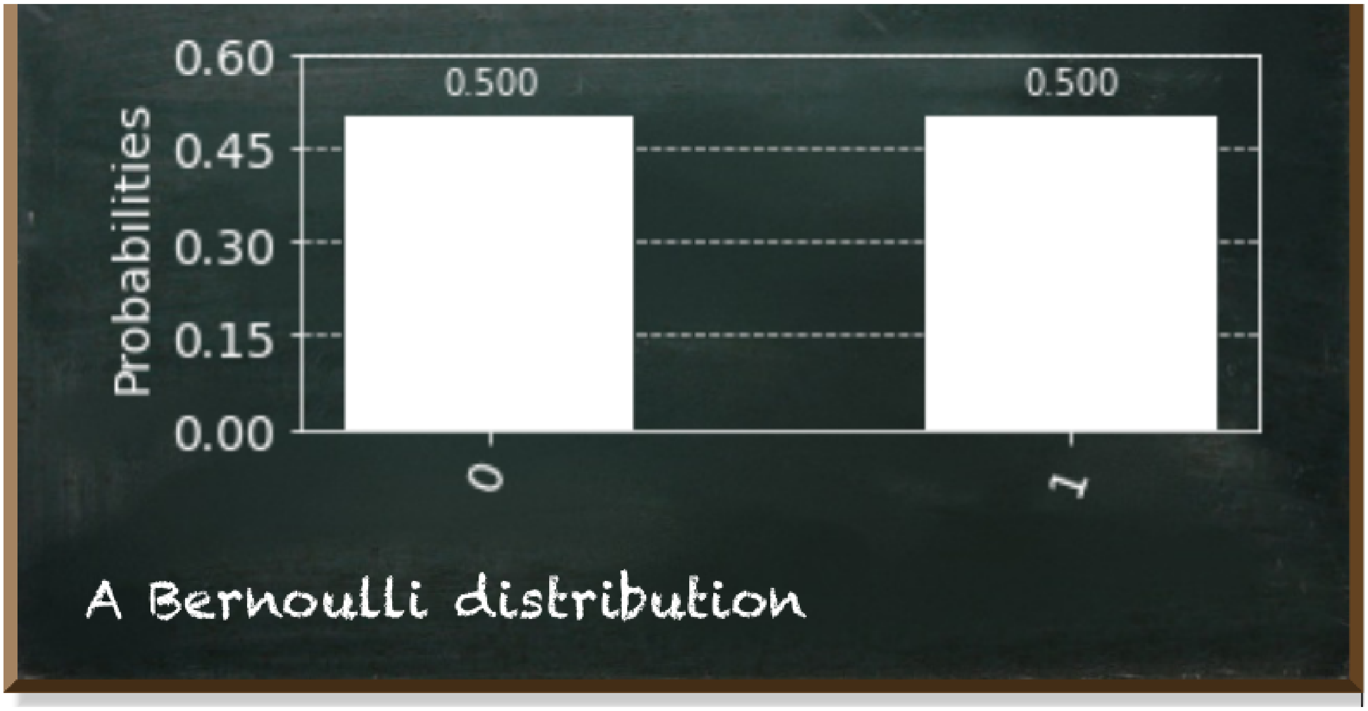
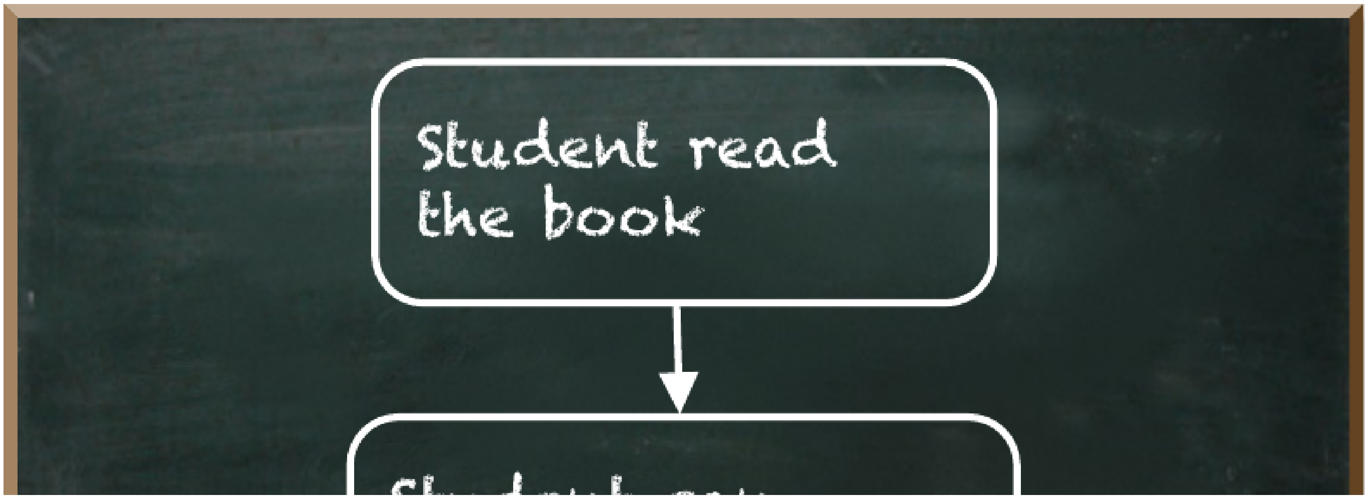


Image by author

In quantum computing, we use the quantum bit (qubit). When you look at it, it is either 0 or 1, each with a distinct probability that depends on the invisible qubit state.

For instance, if the qubit is in a superposition state, such as $|+\rangle$, the chance of measuring it as 0 is 50%. So, the qubit resembles the Bernoulli distribution we just talked about.

Besides variables, a Bayesian network contains conditional dependencies among them. For example, if the student read my book **Hands-On Quantum Machine Learning With Python**, the chance of solving it increases to 90%. Now, we have a two-variable network, as depicted in the following figure.



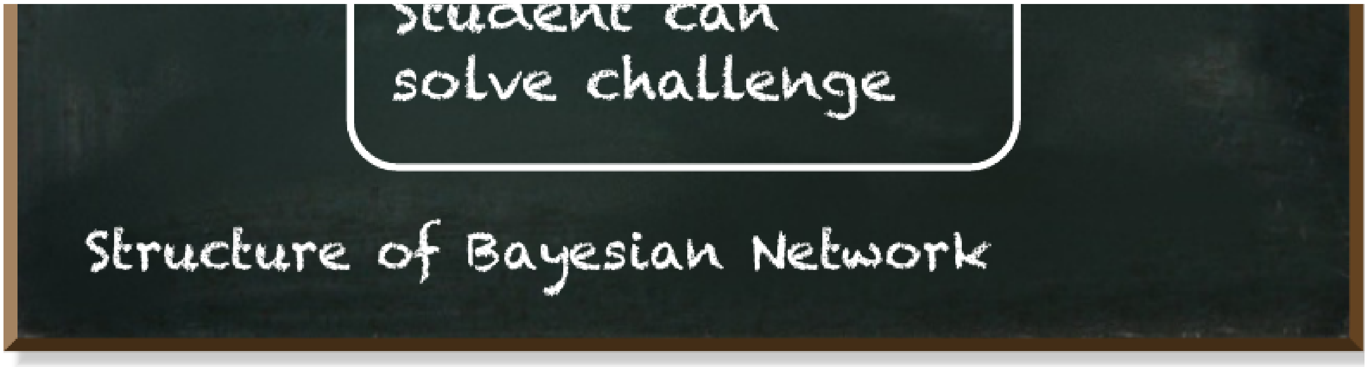


Image by author

In quantum computing, we use entanglement to model dependencies. Entangled qubits are strongly correlated. If we measure a qubit as a certain value, the entangled qubit instantly jumps to a state corresponding to that value. The following image shows the resulting measurement probabilities.

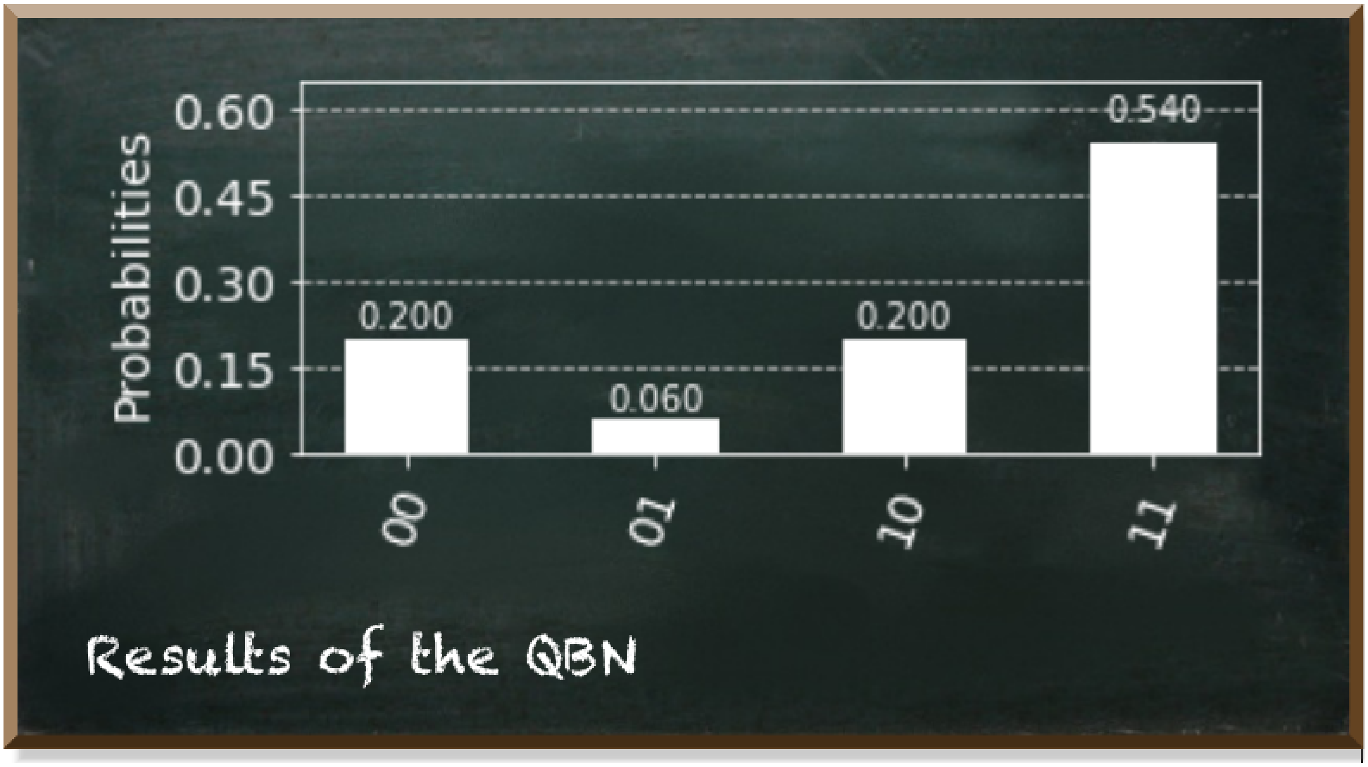


Image by author

The first qubit (number at the top right-hand side) represents whether the student read the book. And, the second (number at the bottom left-hand side) qubit represents whether the student can solve the problem. If we only look at the cases where the first qubit is 1 — that means she read the book — we can see that she solves the problem 90% of the time.

If she didn't read the book, her chances are 50%. In this case, we only look at the states where the first qubit is 0.

Her overall chance to solve the problem is 74%. That is the sum of all states where the second qubit is 1.

Conclusion

Bayesian networks are a simple yet powerful machine learning tool. And, they are handy to get started with quantum machine learning because they share the probabilistic perspective that is paramount in quantum computing.

We experienced how a Bernoulli distribution represents a qubit in superposition. And, we saw that we could use entanglement to model the dependencies between variables in the Bayesian network.

Do you want to learn more about quantum machine learning and quantum Bayesian networks? In my book **Hands-On Quantum Machine Learning With Python**, we build up a QBN from scratch, train it to account for missing values, and use it for inference.

Get the first three chapters for free here.

Quantum Computing Machine Learning Programming Computer Science

Bayesian Machine Learning