

[Show Transcript](#)[Summarize Video](#)

PyTorch optimizers are important tools that help improve how a neural network learns from data by adjusting the model's parameters. By using these optimizers, like stochastic gradient descent (SGD) with momentum or Adam, we can quickly get started learning!

Technical Terms Explained:

Gradients: Directions and amounts by which a function increases most. The parameters can be changed in a direction opposite to the gradient of the loss function in order to reduce the loss.

Learning Rate: This hyperparameter specifies how big the steps are when adjusting the neural network's settings during training. Too big, and you might skip over the best setting; too small, and it'll take a very long time to get there.

Momentum: A technique that helps accelerate the optimizer in the right direction and dampens oscillations.

Quiz Question

In PyTorch, what is the first argument that one passes to `optim.SGD` and `optim.Adam`?

- `lr`, the learning rate
- `momentum`, for momentum
- `model.parameters()`, i.e. the parameters of the model to be optimized

Submit

Code Examples

Assuming `model` is your defined neural network.

`lr=0.01` sets the learning rate to 0.01 for either optimizer.

```
import torch.optim as optim
```

Stochastic Gradient Descent

```
# momentum=0.9 smoothes out updates and can help training
optimizer = optim.SGD(model.parameters(), lr=0.01, momentum=0.9)
```

Adam

```
optimizer = optim.Adam(model.parameters(), lr=0.01)
```

Resources

[PyTorch optimization tutorial](#)

[torch.optim.SGD documentation](#)

[torch.optim.Adam documentation](#)

[Index of PyTorch optimizers](#)