

Session4_CNN

Exercise 2:

ReLU		
3 Epoch loss	5 Epoch loss	10 Epoch loss
Epoch 1/3, Loss: 0.2568046103841238	Epoch 1/5, Loss: 0.27562258411139284	Epoch 1/10, Loss: 0.2562432248802629
Epoch 2/3, Loss: 0.07091887332330238	Epoch 2/5, Loss: 0.07310356413558332	Epoch 2/10, Loss: 0.06814323305916875
Epoch 3/3, Loss: 0.05179131546453424	Epoch 3/5, Loss: 0.051959804337678245	Epoch 3/10, Loss: 0.04756732677291485
	Epoch 4/5, Loss: 0.04138979773811086	Epoch 4/10, Loss: 0.03735966421229375
	Epoch 5/5, Loss: 0.03376727745714866	Epoch 5/10, Loss: 0.03003421639139925
		Epoch 6/10, Loss: 0.025247974069736927
		Epoch 7/10, Loss: 0.020655206115085145
		Epoch 8/10, Loss: 0.01674182915476369
		Epoch 9/10, Loss: 0.013836261996367848
		Epoch 10/10, Loss: 0.012315302737526996
Test Accuracy: 98.55%	Test Accuracy: 98.41%	Test Accuracy: 98.93%

Sigmoid		
3 Epoch loss	5 Epoch loss	10 Epoch loss
Epoch 1/3, Loss: 0.34332016034365526	Epoch 1/5, Loss: 0.33605543517473857	Epoch 1/10, Loss: 0.35223443527966103
Epoch 2/3, Loss: 0.07787811970894239	Epoch 2/5, Loss: 0.0741022059580744	Epoch 2/10, Loss: 0.07766236124067769
Epoch 3/3, Loss: 0.05279775511901349	Epoch 3/5, Loss: 0.05016356805386121	Epoch 3/10, Loss: 0.05212331720953509
	Epoch 4/5, Loss: 0.03779658810548318	Epoch 4/10, Loss: 0.04029387192575257
	Epoch 5/5, Loss: 0.03022139265312692	Epoch 5/10, Loss: 0.031831782074771255
		Epoch 6/10, Loss: 0.02514571268203309
		Epoch 7/10, Loss: 0.02063219449872564
		Epoch 8/10, Loss: 0.016011776153751805
		Epoch 9/10, Loss: 0.012872777331415921
		Epoch 10/10, Loss: 0.009980678269669398
Test Accuracy: 98.65%	Test Accuracy: 98.89%	Test Accuracy: 99.01%

Tanh		
3 Epoch loss	5 Epoch loss	10 Epoch loss
Epoch 1/3, Loss: 0.019956989459438263	Epoch 1/5, Loss: 0.21936564712620366	Epoch 1/10, Loss: 0.24000176493632497
Epoch 2/3, Loss: 0.013464266488144645	Epoch 2/5, Loss: 0.05982722150983968	Epoch 2/10, Loss: 0.05690921085731768
Epoch 3/3, Loss: 0.011222743889537088	Epoch 3/5, Loss: 0.04074680876941966	Epoch 3/10, Loss: 0.038703873884359766
	Epoch 4/5, Loss: 0.03050512055417638	Epoch 4/10, Loss: 0.030178280271550537
	Epoch 5/5, Loss: 0.02339865554827573	Epoch 5/10, Loss: 0.022583907063398113
		Epoch 6/10, Loss: 0.01624009251150974
		Epoch 7/10, Loss: 0.012968596996246045
		Epoch 8/10, Loss: 0.010439835693981341
		Epoch 9/10, Loss: 0.007359307595535612
		Epoch 10/10, Loss: 0.007714346537681377
Test Accuracy: 98.84%	Test Accuracy: 98.85%	Test Accuracy: 99.02%

همانطور که می‌دانیم با افزایش تعداد epoch، مقدار loss ما کاهش می‌یابد چون وزن‌ها بهتر تنظیم شد و احتمال همگرایی بیشتر می‌شود. در نتایج هم این موضوع کاملاً مشخص است.

با تغییر تابع activation به tanh و sigmoid هم می‌توانیم مشاهده کنیم که برای شبکه ما و داده‌های ما tanh بیشتر جواب داده است زیرا:

Rفتار	Range	Activation
صفر برای مقادیر منفی : باعث می شود بعضی نورون ها «مرده» شوند. سریع و پایدار در شبکه های عمیق.	[0, ∞)	ReLU
خروجی صفر- مرکزی- zero: گرادیان ها متعادل تر هستند، یادگیری برای بعضی دیتاستها راحت تر می شود.	[-1, 1]	Tanh

دیتاست MNIST یک دیتاست کوچک و ساده است، اگر دیتاست کوچک باشد، شبکه عمیق زیادی نداشته باشیم و تعداد نورون ها کم باشد، کمک می کند که وزن ها سریع تر و یکنواخت آپدیت شوند.

همچنین داده های ما با ساتفاده از نرمال سازی بین -1 و 1 هستند. این محدوده با Tanh خیلی خوب جور درمی آید، چون خروجی Tanh هم بین -1 و 1 است و اگر وزن ها خیلی کوچک باشند، ReLU ممکن است نورون ها را کمتر فعال کند (Dead neurons)، ولی Tanh با این مقدار کوچک هنوز گرادیان می دهد.

برای شبکه های کوچک و دیتاهای نرمال شده بین -1 و 1، Tanh کاهی بهتر از ReLU عمل می کند. ولی برای شبکه های عمیق و پیچیده یا تصاویر بزرگ، معمولا ReLU بهتر است.

Kernel 5*5 and padding=2		
Relu	Sigmoid	tanh
Epoch 1/3, Loss: 0.20687084111160656	Epoch 1/3, Loss: 0.29967166450041444	Epoch 1/3, Loss: 0.176717336457878
Epoch 2/3, Loss: 0.05652815646872715	Epoch 2/3, Loss: 0.06325990119939054	Epoch 2/3, Loss: 0.047875503613266035

Epoch 3/3, Loss: 0.03984213294343465	Epoch 3/3, Loss: 0.042732165802194716	Epoch 3/3, Loss: 0.032895623230318756
Test Accuracy: 98.47%	Test Accuracy: 98.84%	Test Accuracy: 99.03%

وقتی کرنل را ۵ در ۵ قرار می‌دهیم، برای تابع activation tanh بهتر جواب می‌دهد.

Dropout

یعنی در هر آموزش برخی نورون‌ها خاموش می‌شوند و مانع overfitting می‌شود

شبکه مجبور است با زیرمجموعه‌های مختلفی از نورون‌ها یاد بگیرد و به یک مسیر خاص وابسته نمی‌شود.

```
self.fc1 = nn.Linear(16*8*8, 128)
```

```
self.dropout = nn.Dropout(0.5)
```

```
self.fc2 = nn.Linear(128, 10)
```

Adam to SGD

تغییر adam به sgd نتایج را بسیار بد کرد. SGD فقط یک learning rate ساده دارد

SGD هر وزن را با یک نرخ ثابت آپدیت می‌کند:

$$w = w - \eta \cdot \nabla L$$

این یعنی برای همه‌ی وزن‌ها سرعت یادگیری یکسان است. اما Adam خیلی هوشمند است

برای هر وزن دو چیز را دنبال می‌کند: میانگین گرادیان (m) و میانگین مربعات گرادیان (v) در نتیجه adam بهتر جواب می‌دهد.

تغییر داده

CIFAR10		
Relu	Sigmoid	tanh
Epoch 1/3, Loss: 1.5355850569427472	Epoch 1/3, Loss: 1.5831383744164196	Epoch 1/3, Loss: 1.5343778590716974
Epoch 2/3, Loss: 1.2224273240322348	Epoch 2/3, Loss: 1.2303786232038532	Epoch 2/3, Loss: 1.2227387945060535
Epoch 3/3, Loss: 1.1046153033328483	Epoch 3/3, Loss: 1.0945263263362144	Epoch 3/3, Loss: 1.08586446555984
Test Accuracy: 60.46%	Test Accuracy: 61.47%	Test Accuracy: 62.47%

چون داده‌ی CIFAR10 سخت‌تر و پیچیده‌تر از MNIST است و شبکه‌ی ساده‌ای است، شبکه‌ی ما نتوانست خیلی خوب تشخیص دهد و کار کند. داده CIFAR10 داده‌ی رنگی است و سه کانال دارد درحالی که داده‌ی MNIST یک داده‌ی سیاه و سفید و ساده است.