

به نام خدا



تمرین سری چهارم درس یادگیری عمیق

دکتر محمدی

محمد یارمقدم

96462104

## سوال اول)

در این سوال هدف این است که بررسی شود که بهینه ساز adam در هر شرایطی بهتر از سایر گزینه ها است یا خیر. منطقی آن است که جواب این سوال منفی باشد. چون رقابایی در کنار این بهینه ساز وجود دارد که باید هر چند کم کاربرد هایی داشته باشند که معرفی شده باشند.

بهینه ساز Adam در واقع بهبود یافته بهینه ساز های RMSProp است. این بهینه ساز به سرعت همگرا میشود و مشکل کم شدن اثر learning rate در شبکه را نخواهد داشت. مشکل این روش تنها در هزینه محاسبات صورت گرفته در آن است.

این بهینه ساز از شیب ها و مقدار های قبلی نمونه برای تخمین لحظات اول استفاده می کند. روش بهتر در برخی موارد میتواند Nadam باشد. Nadam همانند Adam است با این تفاوت که مومنتوم را در محاسبات جایگزین شتاب می کند. این روش دقت بیشتری در قسمت داده های اعتبار دارد پس در مواردی که این دقت مورد نظر باشد استفاده از این بهینه ساز توصیه میشود. در کل بهینه ساز های جدید مانند RMSProp, Adadelta, Adam و ... در اغلب شرایط عملکرد مشابه و خوبی دارند و در برخی یوز کیس ها متفاوت میشوند. RMSProp و Adagrad مشابه یکدیگر محسوب میشوند با این تفاوت که learning rate در روش اول نیاز به initialize ندارد و به مرور آپدیت میشود ولی این مقدار در طول زمان اکیدا نزولی آپدیت میشود که باعث آهستگی زمان آموزش میشود و در روش دوم این مقدار با شدت در کم نشده و با سرعت مناسب تری آپدیت میشود. پس در مواردی که پراکندگی داده ها زیاد باشد میتوان از این روش استفاده کرد. هم چنین اگر نرخ ابتدایی learning rate خوب انتخاب نشود، این روش بهتر نتیجه به ما میدهد.

اگر سرعت در آزمایش ما اولویت بالایی داشته باشد، استفاده از adaptive learning rate توصیه میشود. در غیر اینصورت روش های SGD بهتر است. چون با اینکه بیشتر زمان میخواهد اما نتیجه بهتری میدهد و تابع زیان را به حداقل می رساند. استفاده از روش های جدید تر SGD بهتر خواهد بود زیرا همگرایی را راحت تر و با سرعت بیشتری انجام میدهند.

جمع بندی: Adam برای داده های آموزش دقت بالایی دارد و میتوان گفت بهترین گزینه است اما در داده های آزمون و اعتبار دقت را نگه نمی دارد، زیرا سعی در حفظ الگو ها دارد. پس میتوان نتیجه گرفت generalization در این بهینه ساز نرخ بالایی ندارد. در داده هایی با تعداد کم SGD بهتر از adam است. چون دقت تنها در داده های آموزش مهم نیست و در داده های آزمون و اعتبار هم باید دقت حفظ شود.

اما در داده های پراکنده adam بهتر خواهد بود. اگر سرعت نیز برای ما مهم باشد این انتخاب بسیار بهتر خواهد بود. اما در مواردی که دقت در داده های غیر آموزش مهم باشد روش های جدید SGD بهتر است.

لینک های کمکی:

<https://shaoanlu.wordpress.com/2017/05/29/sgd-all-which-one-is-the-best-optimizer-dogs-vs-cats-toy-experiment/>

## سوال سوم)

در این سوال هدف پیاده سازی شبکه CNN معرفی شده با ابزار Pytorch است.

دیتاست CIFAR10 مجموعه داده عکس های وسایل نقلیه و حیوانات است. این مجموعه داده 50000 داده آموزش و 10000 داده تست دارد. این دیتاست پنج batch برای داده های آموزش و یک batch برای داده های تست دارد. در مجموع پس 60000 داده خواهیم داشت. لیبل های این دیتاست عبارت اند از:

- Airplane
- Automobile
- Bird
- Cat
- Deer
- Dog
- Frog
- Horse
- Ship
- Truck

برای پیاده سازی شبکه داده شده بر روی این دیتاست ابتدا داده ها را طبق دستورات داک لود کرده و سپس تعداد batch را مشخص میکنیم. تعداد batch ها از روی اندازه هر batch و تعداد input به دست می آید. سائز کرنل نیز 2 ست شده است.

حال طبق شکل داده شد یک لایه sequential به فرمت داده شده تشکیل می دهیم و سپس یک لایه fully connected طبق شکل تشکیل میدهم. آرگومان های لایه های Conv و Linear شاید بهینه انتخاب نشده باشد اما طبق مطالعات انجام داده شرایط آموزش شبکه و برقراری آن را داراست.

برای بهینه ساز از بهینه ساز adam و برای تابع ضرر CrossEntrphyLoss استفاده کردیم. برای این دو مورد از optim در کتابخانه torch استفاده کردیم و پارامتر های مدل و نرخ یادگیری را برای آن ست کردیم.

سپس در تابع forward این دو لایه را به ترتیب فراخوانی میکنیم و سپس نتیجه لایه آخر را برمیگردانیم.

برای آموزش این شبکه Epoch 2 در نظر گرفتیم و در ابتدا مقادیر بهینه ساز را صفر می کنیم تا برای هر دسته مقادیر جداگانه محاسبه شود. سپس خروجی مدل و شبکه را در حالت forward محاسبه می کنیم و در نهایت احتمال خروجی را در لیبل های مدل حساب کرده و ببینیم آنها به عنوان لیبل معرفی می کنیم. در نهایت نیز از روی خطای مدل در عملیات back propagation وزن ها را آپدیت میکنیم.

طبق مقادیر خروجی که در نوت بوک موجود است می توان دریافت که خطا در هر دسته کاهش و دقت افزایش یافته است. اما دقت مناسبی را مدل ندارد. علت را میتوان در تعداد epoch کم دانست. در کل ران شدن مدل زمان بر بود و علت این کم بودن احتمالا این موضوع است.

در آخر داده های تست را در مدل fit کردیم تا دقت مدل را در لیبل های مختلف را ببینیم. برای هر داده تست خروجی مدل را با خروجی اصلی مقایسه می کنیم و تعداد حدس درست را بر کل تقسیم می کنیم تا دقت در هر کلاس محاسبه شود.

---

Accuracy of the network on the 10000 test images: 47 %

Accuracy of plan is: 49.500 %  
Accuracy of car is: 55.100 %  
Accuracy of bird is: 16.700 %  
Accuracy of cat is: 26.200 %  
Accuracy of deer is: 40.500 %  
Accuracy of dog is: 58.400 %  
Accuracy of frog is: 55.500 %  
Accuracy of horse is: 48.800 %  
Accuracy of ship is: 54.300 %  
Accuracy of truck is: 67.300 %

## سوال دوم

علاوه بر حل تشریحی و روی کاغذ این مورد کد نیز برای تسلط بیشتر زده شد و بر روی نوت بوک موجود است.

### سوال دو

ابتدا مقادیر ورودی‌های موجود در شبکه را محاسبه می‌کنیم:

$$h_1 = i_1 w_{11} + i_2 w_{12} = 0.2 \times 2 + 0.3 \times 0.5 = 1.9$$

فرض: مقدار اولیه وزن‌ها

$$w_{11} = 0.2 \quad w_{12} = 0.3$$

$$h_2 = i_1 w_{21} + i_2 w_{22} = 3 \times 0.2 + 0.5 \times 0.2 = 2.1$$

$$w_{21} = 0.3 \quad w_{22} = 0.2$$

$$z = h_1 w_{31} + h_2 w_{32} = 1.9 \times 0.2 + 2.1 \times 0.5 = 4.1$$

$$i_1 = 3 \quad w_{31} = 0.2$$

$$i_2 = 0.5 \quad w_{32} = 0.5$$

$$a = \sigma(z) = \sigma(h_1 w_{31} + h_2 w_{32}) = \sigma(1.9 \times 0.2 + 2.1 \times 0.5)$$

$$= \sigma(4.1) = 0.98$$

$$j = \frac{(0 - y)^2}{2} = \frac{(0.98 - 1)^2}{2} = 0.0001$$

$$w_{\text{new}} = w_{\text{old}} - \alpha \left( \frac{\partial J}{\partial w} \right) \quad \text{طبق رابطه آیینیت شبکه دایره}$$

سپس باید برای هر وزن (ضریب) به روش زیر عمل کنیم:

$$\frac{\partial J}{\partial w_{11}} = \frac{\partial J}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w_{11}} = \sigma(z) \times (1 - \sigma(z)) \times (0 - y) \times h_2 = 0.98 \times (1 - 0.98) \times (-0.02) \times 2.1 = -0.0004$$

$$\frac{\partial J}{\partial w_{12}} = (0 - y) \times (\sigma(z) \times (1 - \sigma(z))) \times h_1 = (-0.02) \times (0.98 \times 0.02) \times 1.9 = -0.0007$$

$$\frac{\partial J}{\partial w_{21}} = (0 - y) \times (\sigma(z) \times (1 - \sigma(z))) \times i_1 = (-0.02) \times (0.98 \times 0.02) \times 3 = -0.0012$$

$$\frac{\partial J}{\partial w_{22}} = (0 - y) \times (\sigma(z) \times (1 - \sigma(z))) \times i_2 = (-0.02) \times (0.98 \times 0.02) \times 0.5 = -0.0002$$

$$\frac{\partial J}{\partial w_f} = (0-y) \times (h_1 \times g(z) \times (1-g(z))) \times \hat{z}_f = (-.12) \times (.19 \times .187 \times .12) \times \omega$$

$$= -.014$$

$$\frac{\partial J}{\partial w_i} = (0-y) \times (h_1 \times g(z) \times (1-g(z))) \times \hat{z}_i = (-.12) \times (.19 \times .187 \times .12) \times \omega$$

$$= -.019$$

$$w_{y_n} = w_{y_0} - \alpha \frac{\partial J}{\partial w_y} = .15 - .1 \times (-.014) = .1514$$

$$w_{\omega_n} = w_{\omega_0} - \alpha \frac{\partial J}{\partial w_{\omega}} = .12 - .1 \times (-.014) = .1214$$

$$w_{\Sigma_n} = w_{\Sigma_0} - \alpha \frac{\partial J}{\partial w_{\Sigma}} = .12 - .1 \times (-.1180) = .1318$$

$$w_{\tau_n} = w_{\tau_0} - \alpha \frac{\partial J}{\partial w_{\tau}} = .12 - .1 \times (-.111) = .1311$$

$$w_{\tau_n} = w_{\tau_0} - \alpha \frac{\partial J}{\partial w_{\tau}} = .13 - .1 \times (-.014) = .1314$$

$$w_{i_n} = w_{i_0} - \alpha \frac{\partial J}{\partial w_i} = .12 - .1 \times (-.019) = .1219$$

كل الـ epoch (تكرار) . هـي مرحلة تدريب الـ epoch (تكرار) .

$$h_1 = w_i \times i_1 + w_{\tau} \times i_{\tau} = .1219 \times 3 + .1311 \times \omega = 1.101$$

$$h_{\tau} = w_{\tau} \times i_1 + w_{\Sigma} \times i_{\tau} = .1311 \times 3 + .1318 \times \omega = 1.201$$

$$Z = h_1 + h_{\tau} = 1.101 + 1.101 = 2.202$$

$$O = g(Z) = g(2.202) = .99$$

$$J = \frac{(0-y)^2}{2} = \frac{(.99-1)^2}{2} = .00005$$

$$\frac{\partial J}{\partial w_y} = (0-y) \times (0.2) \times (1-0.2) \times h_r = (-0.1) \times (0.99 \times 0.1 \times 2.208) \\ = -0.0021$$

$$\frac{\partial J}{\partial w_o} = (0-y) \times (0.2) \times (1-0.2) \times h_i = (-0.1) \times (0.99 \times 0.1) \times 2.108 \\ = -0.00208$$

$$\frac{\partial J}{\partial w_x} = (0-y) \times (h_r \times 0.2) \times (1-0.2) \times i_r = (-0.1) \times (2.208 \times 0.99 \times 0.1 \times 0.1 \times 0.2) \\ = -0.00042$$

$$\frac{\partial J}{\partial w_r} = (0-y) \times (h_r \times 0.2) \times (1-0.2) \times i_i = (-0.1) \times (2.208 \times 0.99 \times 0.1) \\ = -0.0043$$

$$\frac{\partial J}{\partial w_r} = (0-y) \times (h_i \times 0.2) \times (1-0.2) \times i_r = (-0.1) \times (2.108 \times 0.99 \times 0.1) \\ \times 0.2 = -0.0042$$

$$\frac{\partial J}{\partial w_i} = (0-y) \times (h_i \times 0.2) \times (1-0.2) \times i_i = (-0.1) \times (2.108 \times 0.99 \times 0.1) \\ \times 0.2 = -0.0042$$

$$w_{y_n} = w_{y_0} - \alpha \frac{\partial J}{\partial w_y} = 0.1 + 0.1 \times 0.0021 = 0.1002$$

$$w_{o_n} = w_{o_0} - \alpha \frac{\partial J}{\partial w_o} = 0.1 + 0.1 \times 0.00208 = 0.100208$$

$$w_{x_n} = w_{x_0} - \alpha \frac{\partial J}{\partial w_x} = 0.1 + 0.1 \times 0.00042 = 0.10042$$

$$w_{r_n} = w_{r_0} - \alpha \frac{\partial J}{\partial w_r} = 0.1 + 0.1 \times 0.0043 = 0.10043$$

$$w_{r_n} = w_{r_0} - \alpha \frac{\partial J}{\partial w_r} = 0.1 + 0.1 \times 0.0042 = 0.10042$$

$$w_{i_n} = w_{i_0} - \alpha \frac{\partial J}{\partial w_i} = 0.1 + 0.1 \times 0.0042 = 0.10042$$

مقدار خطای موزن کاهش است پس شبکه در حال تعادل است. در ابتدا نیز کاهش می‌یافت.