به نام خداوند جان و خرد

نیمسال دوم 1401-1402

استاد درس: دکتر منصور رزقی آهق

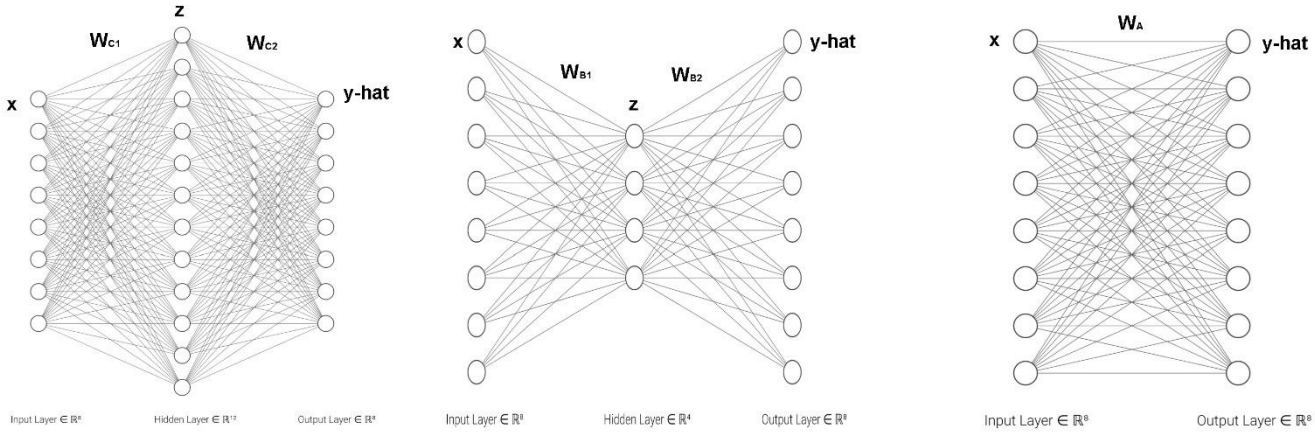
دانشجو: بهار مهدوی
40152521337

Deep Learning
Homework-2

Tarbiat Modares University

تمرین 1:

فرض کنید در سه شبکه زیر تابع هدف یکسان بوده و تمامی توابع فعال ساز خطی باشند. تعداد نرون‌های لایه میانی شبکه (B) برابر با ۴ و در شبکه (C) برابر با ۱۲ می باشد. اگر (x_i, y_i) داده‌های آموزشی و $\hat{y}_i$ جواب تقریبی این شبکه باشد، از نظر منطقی کیفیت جواب های شبکه ها را مقایسه و اثبات ریاضی آن را بنویسید. همچنین اگر شرطی در مورد تعداد داده ها هست، بیان کنید.



شبکه (C) شبکه (B) شبکه (A)

شبکه A:

در شبکه A، 8 نورون در لایه ورودی و 8 نورون در لایه خروجی وجود دارد که اگر بر طبق تعداد فیچرهای لایه خروجی، فرض کنیم linear classification هشت کلاسه داریم، اگر x_i عضو کلاس C_j باشد y_i ما یک بردار 8 تایی خواهد بود که j امین عنصر آن یک و مابقی آن صفر است. $\hat{y}_i$ برابر حاصل ضرب پارامتر w در x است که پارامترهای w عضو یک ماتریس 8*8 است. بنابراین در ماتریسی که 8 سطر و 8 ستون دارد حداکثر Rank، 8 میشود:

$$y(x_i, y_i) \quad if \quad x_i \in C_j \qquad y_i = \begin{pmatrix} 0 \\ \dots \\ j \\ \dots \\ 0 \end{pmatrix} \qquad \hat{y}_i = \begin{pmatrix} w_1^T \\ w_2^T \\ \dots \\ w_8^T \end{pmatrix} x \;\rightarrow\; \hat{y}_i = w_A x \quad w_A \in R^{8*8} \quad Rank(w_A) \leq 8$$

$$w_A = \{w_A | Rank(w_A) \leq 8 \quad w_A \in R^{8*8}\}$$

شبکه B:

اگر به شبکه A یک لایه پنهان $(x \in R^4)$ اضافه گردد شبکه B بدست می‌آید. در این صورت همانطور که میبینیم حداکثر Rank، 4 میشود:

$$\hat{y}_i = w_{B2} z \quad w_{B2} \in R^{8*4} \quad Rank(w_{B2}) \leq 4$$
$$z = w_{B1} x \quad w_{B1} \in R^{4*8} \quad Rank(w_{B1}) \leq 4$$
$$\hat{y}_i = w_{B2} w_{B1} x \quad if \quad w_{B2} w_{B1} = \overline{w_B} \quad \overline{w_B} \in R^{8*8} \quad Rank(\overline{w_B}) \leq 4$$

$$w_B = \{\overline{w_B} | \overline{w_B} = w_{B2} w_{B1} \quad w_{B1} \in R^{4*8} \;\&\; w_{B2} \in R^{8*4}\}$$

Rank4 نمیتواند Rank8 را پوشش دهد. به این ترتیب حتی در بهینه ترین شرایط شبکه B، قدرت کمتری نسبت به شبکه A و C خواهد داشت.

شبکه C:

اگر به شبکه A یک لایه پنهان $(x \in R^{12})$ اضافه گردد شبکه C بدست می‌آید. در این صورت هم همانطور که میبینیم حداکثر Rank، 8 میشود:

$$\hat{y}_\iota = w_{C2} z \quad w_{C2} \in R^{8*12} \quad Rank(w_{C2}) \leq 8$$
$$z = w_{C1} x \quad w_{C1} \in R^{12*8} \quad Rank(w_{C1}) \leq 8$$
$$\hat{y}_\iota = w_{C2} w_{C1} x \quad if \quad w_{C2} w_{C1} = \overline{w_C} \quad \overline{w_C} \in R^{8*8} \quad Rank(\overline{w_C}) \leq 8$$

$$w_C = \{\overline{w_C} | \overline{w_C} = w_{C2} w_{C1} \quad w_{C1} \in R^{12*8} \ \& \ w_{C2} \in R^{8*12}\}$$

پس در بهینه ترین شرایط شبکه C، برای داده های train مشابه شبکه A جواب خواهد داد. ولی چون در مسئله، Active function خطی در نظر گرفته شده است و چون با عمیق کردن شبکه تعداد پارامترها افزایش یافته است، در صورتی که تعداد داده ها کافی نباشد، برای داده های test برای شبکه C احتمال Over fiting مطرح خواهد شد و این امکان وجود دارد که حتی نتیجه شبکه C را در مقایسه با شبکه A خرابتر کند. اگر Active function، غیر خطی در نظر گرفته میشد شرایط عوض میشد و عمیق تر شدن شبکه میتوانست در داده های test نتیجه بهتری را ایجاد نماید.

پس به طور خلاصه میشه ثابت کرد که w_B زیر مجموعه w_C است و w_C با w_A برابر است:

$$w_B \subset w_C = w_A$$

تمرین 2:

الف) اگر $p(y|x) = Lap\ (y|\hat{y}(x, w), \sigma I)$ را از توزیع لاپلاس فرض کنیم، چه تابع هدفی بدست می آید؟

با فرض اینکه خطای بین y و $\hat{y}$ از توزیع لاپلاس با میانگین صفر و واریانس σI پیروی میکند، خواهیم داشت:

$$y - \hat{y} \sim Lap(0, \sigma I) \quad \rightarrow \quad p(y_i|x_i) \triangleq Lap\ (y|\hat{y}(x, w), \sigma I)$$

برای به دســـت آوردن Maximum likelihood Estimation (MLE) برای توزیع لاپلاس، باید مقدار $\hat{y}(x, w)$ را پیدا کنیم که تابع likelihood را max کند به این ترتیب با استفاده از MLE خواهیم داشت:

$$max \prod_{i=1}^{m} p(y_i|x_i) \sim Lap\ (y|\hat{y}(x, w), \sigma I)$$

$$Lap\ (y|\hat{y}(x, w), \sigma I) = \frac{1}{2\sigma I} e^{\left(-\frac{|y_i - \hat{y}_i|}{\sigma I}\right)} \quad \rightarrow \quad max \prod_{i=1}^{m} \frac{1}{2\sigma I} e^{\left(-\frac{|y_i - \hat{y}_i|}{\sigma I}\right)}$$

لگاریتم طبیعی می‌گیریم و خواهیم داشت:

$$arg\ max\ ln \prod_{i=1}^{m} \frac{1}{2\sigma I} e^{\left(-\frac{|y_i - \hat{y}_i|}{\sigma I}\right)} \quad \rightarrow \quad arg\ max \sum_{i=1}^{m} ln \frac{1}{2\sigma I} e^{\left(-\frac{|y_i - \hat{y}_i|}{\sigma I}\right)}$$

$$arg\ max \sum_{i=1}^{m} ln \frac{1}{2\sigma I} + \sum_{i=1}^{m} lne^{\left(-\frac{|y_i - \hat{y}_i|}{\sigma I}\right)} \quad \rightarrow \quad mln \frac{1}{2\sigma I} + \sum_{i=1}^{m} -\frac{|y_i - \hat{y}_i|}{\sigma I}$$

$$\xrightarrow{\sigma I>0} arg\ max - \sum_{i=1}^{m} |y_i - \hat{y}_i| \ \rightarrow \ arg\ min \sum_{i=1}^{m} |y_i - \hat{y}_i(x,w)|$$

Max منفی مجموع خطاهای مطلق، معادل Min مجموع خطاهای مطلق یا Norm 1 Loss (Mean Absolute Error) است. به این ترتیب مطابق اثبات بالا، با استفاده از Maximum likelihood Estimation از توزیع لاپلاس تابع هدف Norm 1 بدست خواهد آمد.

ب) با چه فرضی بروی p(y|x) تابع هدف مربوط به رگرسیون لاجستیک (Cross-Entropy) بدست می آید؟

اگر فرض کنیم خطای بین y و $\hat{y}$ از توزیع برنولی پیروی می‌کند، خواهیم داشت:

$$y - \hat{y} \sim Ber \ \rightarrow \ p(y_i|x_i) \triangleq Ber\ (y|\hat{y}(x,w))$$

برای به دســـت آوردن Maximum likelihood Estimation (MLE) برای توزیع برنولی، باید مقدار $\hat{y}(x,w)$ را پیدا کنیم که تابع likelihood را max کند به این ترتیب با استفاده از MLE خواهیم داشت:

$$max \prod_{i=1}^{m} p(y_i|x_i) \sim Ber\ (y|\hat{y}(x,w))$$

$$p(y|x,w) = \hat{y}(x,w)^y \big(1 - \hat{y}(x,w)\big)^{1-y} \quad y = 0 \ \ or \ \ y = 1, \ \ 0 \le \hat{y} \le 1 \ \rightarrow \ max \prod_{i=1}^{m} \hat{y}^y\ (1-\hat{y})^{1-y}$$

لگاریتم می‌گیریم و خواهیم داشت:

$$arg\ max\ log \prod_{i=1}^{m} \hat{y}^y\ (1-\hat{y})^{1-y} \ \rightarrow \ \sum_{i=1}^{m} log\ \hat{y}^y (1-\hat{y})^{1-y}$$

$$arg\ max \sum_{i=1}^{m} y\ log\ \hat{y} + (1-y) log(1-\hat{y})$$

به این ترتیب مطابق اثبات بالا، با استفاده از Maximum likelihood Estimation با فرض توزیع برنولی بر روی $p(y_i|x_i)$ تابع هدف لاجستیک رگرسیون دو کلاسه (Cross-Entropy) بدست می‌آید.

تمرین 3:

شـــبکه های زیر را بر روی مجموعه داده IRIS، با 100 تکرار و تابع هدف Cross-Entropy آموزش دهید. مقدار دقت و خطا را برای داده های آموزشی و تست را برای هر تکرار گزارش دهید.

1. یک شبکه بدون لایه پنهان
2. یک شبکه با سه لایه پنهان و 24 نرون برای هر لایه با تابع فعالساز خطی
3. یک شبکه با سه لایه پنهان و 24 نرون برای هر لایه با تابع فعالساز ReLU

neural network model	a network without hidden layers		a network with 3 hidden layers 24 neurons per layer linear activation function		a network with 3 hidden layers 24 neurons per layer the ReLU activation function	
optimizer	SGD optimizer	Adam optimizer	SGD optimizer	Adam optimizer	SGD optimizer	Adam optimizer
Accuracy	76%	66%	90%	90%	83%	90%
Cross Entropy Loss Function	Epoch: 0 Loss: 1.1130679845809937 Epoch: 10 Loss: 1.0911153554916382 Epoch: 20 Loss: 1.0517007112503052 Epoch: 30 Loss: 1.0112653970718384 Epoch: 40 Loss: 0.9756056070327759 Epoch: 50 Loss: 0.9460235238075256 Epoch: 60 Loss: 0.9221143126487732 Epoch: 70 Loss: 0.9029256701469421 Epoch: 80 Loss: 0.8874469995498657 Epoch: 90 Loss: 0.874805212020874	Epoch: 0 Loss: 1.139886736869812 Epoch: 10 Loss: 1.0672699213027954 Epoch: 20 Loss: 1.0029655694961548 Epoch: 30 Loss: 0.9499607086181641 Epoch: 40 Loss: 0.9092450737953186 Epoch: 50 Loss: 0.878618061542511 Epoch: 60 Loss: 0.8551142811775208 Epoch: 70 Loss: 0.8366459012031555 Epoch: 80 Loss: 0.8217170238494873 Epoch: 90 Loss: 0.8093350529670715	Epoch: 0 Loss: 1.080617785453796 Epoch: 10 Loss: 0.806310355663299 Epoch: 20 Loss: 0.485130846500396 Epoch: 30 Loss: 0.356783837080001 Epoch: 40 Loss: 0.280783057212829 Epoch: 50 Loss: 0.235078573226928 Epoch: 60 Loss: 0.199553579092025 Epoch: 70 Loss: 0.165643408894538 Epoch: 80 Loss: 0.133510872721672 Epoch: 90 Loss: 0.106084622442722	Epoch: 0 Loss: 1.1102420091629028 Epoch: 10 Loss: 0.23713721334934235 Epoch: 20 Loss: 0.06534051150083542 Epoch: 30 Loss: 0.01790588162839412 Epoch: 40 Loss: 0.00970490090548992 Epoch: 50 Loss: 0.00708628958091139 Epoch: 60 Loss: 0.00493485899642109 Epoch: 70 Loss: 0.00373480259440839 Epoch: 80 Loss: 0.00292301503941416 Epoch: 90 Loss: 0.00232839281670749	Epoch: 0 Loss: 1.102975130081178 Epoch: 10 Loss: 1.057321190834045 Epoch: 20 Loss: 0.959731519222259 Epoch: 30 Loss: 0.810281217098236 Epoch: 40 Loss: 0.675509452819824 Epoch: 50 Loss: 0.585866928100585 Epoch: 60 Loss: 0.513817369937896 Epoch: 70 Loss: 0.456795364618301 Epoch: 80 Loss: 0.409913271665573 Epoch: 90 Loss: 0.366027265787124	Epoch: 0 Loss: 1.1149022579193115 Epoch: 10 Loss: 0.5995610356330872 Epoch: 20 Loss: 0.31550055742263794 Epoch: 30 Loss: 0.10892707854509354 Epoch: 40 Loss: 0.03076659515500068 Epoch: 50 Loss: 0.01506648678332567 Epoch: 60 Loss: 0.00915161240845918 Epoch: 70 Loss: 0.00672238925471901 Epoch: 80 Loss: 0.00501444796100258 Epoch: 90 Loss: 0.00380288017913699

نتایج نشـــان می‌دهد که اگر تعداد لایه ها زیادتر و شـــبکه عمیق تر گردد، با افزایش تعداد پارامترها performancy مدل ما برای داده های train افزایش میابد. بدین معنی که $\hat{y}$ پیشـبینی شـده‌ای که از output شـبکه بدسـت می‌آید به y حقیقی داده ها نزدیکتر شـده و در نتیجه میزان loss کاهش بیشـتری یافته و به صـفر نزدیکتر می‌گردد.

اما این نتیجه ممکن است برای داده های test متفاوت باشد و در شبکه عمیقی که از linear active function استفاده میکند تا یه حدی افزایش تعداد لایه ها و نورون‌ها (پارامترها) به بهبود accuracy کمک کند ولی از یه ســطحی به بعد (اگرچه برای داده های train نتیجه خیلی خوبی نشـــان می‌دهد) به علت over fitting دقت مدل برای داده های test به شدت کاهش یابد. (به عنوان مثال در شبکه دوم، افزایش تعداد نورون‌های لایه‌های پنهان تا 100 عدد همچنان accuracy شـــبکه را افزایش داد ولی انتخاب 1000 نورون برای هر لایه accuracy را تا میزان 30% کاهش داد!) این نتیجه برای شـــبکه عمیقی که از non-linear active function مثل تابع فعالساز ReLU استفاده میکند کاملا متفاوت است و افزایش تعداد لایه‌ها و یا تعداد نورون‌ها علاوه بر اینکه performancy مدل ما را برای داده های train افزایش می‌دهد، accuracy داده های test را نیز بهبود می‌دهد. (در مثال قبل برای شبکه سوم، افزایش تعداد نورون‌های لایه‌های پنهان تا 100 یا 1000 عدد همچنان accuracy شبکه را افزایش داد و کاهشی در عملکرد مدل نه برای داده های train و نه برای test مشاهده نگردید.)

همچنین استفاده از روش بهینه سازی Adam در مقایسه با SGD برای هر سه مدل تاثیر به سزایی در کاهش loss و افزایش performancy مدل ما برای داده های train داشت.

کد مرتبط با تمرین 3:

- DL-HW2-Q3-Mahdavi.ipynb

نکته: برای حل سوال 3، از کتابخانه پایتورچ (Pytorch) استفاده کنید.

منابع آموزشی:

1. Simple Iris Dataset Classification Using Pytorch
2. Pytorch Tutorials