

딥러닝 신경망 모형

1. 딥러닝 신경망 개요

1. 딥러닝 신경망개요

[1] 딥러닝 신경망이란

1. 딥러닝 신경망은 인공지능의 한 분야로, 인공 신경망을 이용하여 데이터를 학습하고 예측하는 기술이다.
2. 이 신경망은 인간의 뇌 구조와 유사한 계층적인 모델로 구성되어 있으며, 여러 층의 노드(뉴런)들이 서로 연결되어 정보를 처리한다.

[2] 신경망의 기본 구조

1. 입력층(Input Layer)

원시 데이터를 받아들인다.

데이터는 입력층의 각 노드로 전달되며, 각 노드는 하나의 입력 특징을 나타낸다.

1. 딥러닝 신경망개요

2. 은닉층(Hidden Layers)

입력 데이터를 점차적으로 처리한다.

여러 개의 은닉층이 존재할 수 있으며, 각 은닉층은 이전 층의 출력을 받아 새로운 출력을 생성한다.

은닉층의 노드는 가중치(weight)와 편향(bias)을 가지며, 활성화 함수(activation function)를 통해 출력을 계산한다.

3. 출력층(Output Layer)

최종 결과를 출력 한다.

출력층의 형태는 해결하려는 문제의 유형에 따라 달라질 수 있다
(예: 회귀 문제에서는 연속적인 값, 분류 문제에서는 클래스 레이블).

[3] 활성화 함수(Activation Function)

- 활성화 함수는 각 노드의 출력을 결정하는 비선형 함수로, 신경망의 비선형성을 부여한다.
- 대표적인 활성화 함수로는 시그모이드(sigmoid), 탄젠트 하이퍼볼릭(tanh), 렐루(ReLU), 소프트맥스(softmax) 등이 있다.

[4] 학습 과정

• 전방 전달(Forward Propagation)

입력 데이터를 신경망의 각 층을 통해 전달하여 최종 출력을 계산한다.

• 오차 계산(Error Calculation)

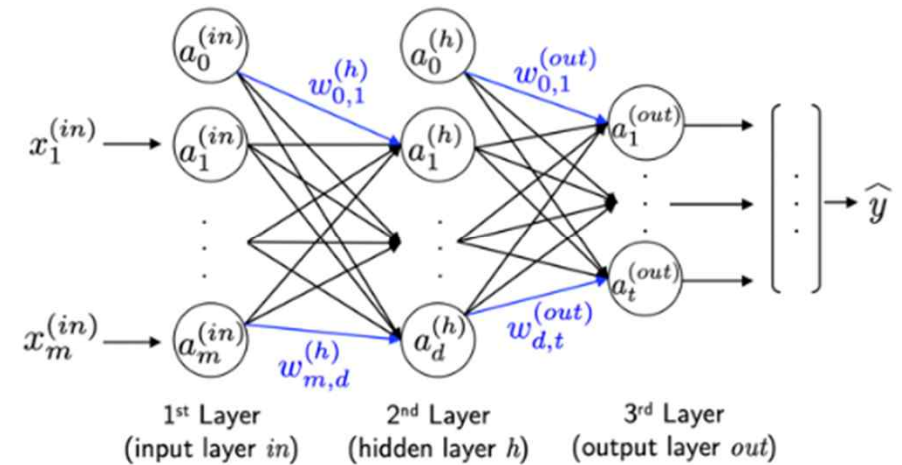
출력층에서 실제값과 예측값 간의 오차를 계산한다.

대표적인 오차 함수로는 평균 제곱 오차(MSE), 교차 엔트로피 손실(Cross-Entropy Loss) 등이 있다.

1. 딥러닝 신경망개요

$$z_1^{(h)} = a_0^{(in)} w_{0,1}^{(h)} + a_1^{(in)} w_{1,1}^{(h)} + \cdots + a_m^{(in)} w_{m,1}^{(h)}$$

$$a_1^{(h)} = \phi(z_1^{(h)})$$



• 역전파(Backpropagation)

오차를 기반으로 가중치와 편향을 조정하여 신경망을 학습시킨다.
오차가 각 층을 거슬러 올라가며 가중치의 기울기를 계산하고, 이를
통해 가중치를 업데이트한다.

[5] 딥러닝 신경망의 종류

- **컨볼루션 신경망(CNN)**

주로 이미지 인식에 사용되며, 컨볼루션 층과 풀링 층을 통해 특징을 추출한다.

- **순환 신경망(RNN)**

시계열 데이터 처리에 특화되어 있으며, 이전 시간 단계의 정보를 기억한다.

LSTM(Long Short-Term Memory)와 GRU(Gated Recurrent Unit) 같은 변형된 구조를 사용하여 긴 의존 관계를 처리한다.

1. 딥러닝 신경망개요

[6] 딥러닝 신경망의 적용 분야

1. 이미지 인식

CNN을 이용하여 이미지 분류, 객체 검출, 얼굴 인식 등 다양한 응용 분야에 사용된다.

2. 음성 인식

음성 데이터를 텍스트로 변환하거나, 음성을 기반으로 명령을 인식하는 시스템에 적용된다.

3. 자연어 처리(NLP)

문서 분류, 기계 번역, 감정 분석 등 텍스트 데이터를 처리하는 데 사용된다.

4. 자율 주행

주변 환경을 인식하고, 주행 경로를 결정하는 데 딥러닝 신경망이 사용된다.

[7] 딥러닝의 발전

- 컴퓨팅 파워의 증가와 대량의 데이터 접근성 향상 덕분에 딥러닝 신경망의 발전이 가능하게 되었으며, 앞으로도 계속해서 발전할 가능성이 높다.

2. 딥러닝 모델 구현 실습


2. 딥러닝 모델 구현 실습



Boston_train.csv 데이터 파일: 선형 회귀 모델

Boston Train					↓ ↗
	CRIM	ZN	INDUS	NOX	
1	2.3004	0.0	19.58	0.605	
2	13.3598	0.0	18.1	0.693	
3	0.12744	0.0	6.91	0.448	
4	0.15876	0.0	10.81	0.413	
5	0.03768	80.0	1.52	0.404	
6	0.03705	20.0	3.33	0.4429	

2. 딥러닝 모델 구현 실습

 boston_train.csv
스프레드시트

 boston_test.csv
스프레드시트

 boston_train 파일로 텐서플로 모델 만들고 boston_test파일로 평가지표를 구해줘 

boston_train 파일로 텐서플로 모델 만들고 boston_test파일로 평가지표를 구해줘




이제 보스턴 주택 데이터셋을 사용하여 딥러닝 모델을 만들고, 평가 지표를 계산해보겠습니다. 주어진 데이터셋을 사용하여 회귀 모델을 생성하겠습니다.

1. 데이터 준비 및 전처리

먼저 데이터셋을 텐서플로우에서 사용할 수 있도록 준비합니다.

python

 코드 복사

```
import pandas as pd
import tensorflow as tf
from sklearn.preprocessing import StandardScaler

# 데이터 로드
train_file_path = '/mnt/data/boston_train.csv'
```

ChatGPT 현재 버전에서는 딥러닝 모델을 실행시키지 못하므로 구글 코랩으로 소스를 복사하여 실행시킨다

2. 딥러닝 모델 구현 실습

Breast Cancer.csv 데이터 파일: 2진 분류 모델

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE		
1	mean radi	mean text	mean peri	mean are	mean smc	mean con	mean con	mean con	mean con	mean sym	mean frac	radius err	texture en	perimeter	area error	smoothne	compactn	concavity	concave	symmetry	fractal	din	worst radi	worst text	worst peri	worst are	worst smc	worst con	worst con	worst con	worst sym	worst frac	target
2	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.0064	0.04904	0.05373	0.01587	0.03003	0.00619	25.38	17.33	184.6	2019	0.1622	0.6656	0.7119	0.2654	0.4601	0.1189	0		
3	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.00523	0.01308	0.0186	0.0134	0.01389	0.00353	24.99	23.41	158.8	1956	0.1238	0.1866	0.2416	0.186	0.275	0.08902	0		
4	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.00457	23.57	25.53	152.5	1709	0.1444	0.4245	0.4504	0.243	0.3613	0.08758	0		
5	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.00921	14.91	26.5	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.173	0		
6	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149	0.02461	0.05688	0.01885	0.01756	0.00512	22.54	16.67	152.2	1575	0.1374	0.205	0.4	0.1625	0.2364	0.07678	0		
7	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.00751	0.03345	0.03672	0.01137	0.02165	0.00508	15.47	23.75	103.4	741.6	0.1791	0.5249	0.5355	0.1741	0.3985	0.1244	0		
8	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.00431	0.01382	0.02254	0.01039	0.01369	0.00218	22.88	27.66	153.2	1606	0.1442	0.2576	0.3784	0.1932	0.3063	0.08368	0		
9	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856	50.96	0.00881	0.03029	0.02488	0.01448	0.01486	0.00541	17.06	28.14	110.6	897	0.1654	0.3682	0.2678	0.1556	0.3196	0.1151	0		
10	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.00573	0.03502	0.03553	0.01226	0.02143	0.00375	15.49	30.73	106.2	739.3	0.1703	0.5401	0.539	0.206	0.4378	0.1072	0		
11	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.00715	0.07217	0.07743	0.01432	0.01789	0.01008	15.09	40.68	97.65	711.4	0.1853	1.058	1.105	0.221	0.4366	0.2075	0		
12	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.00403	0.00927	0.01101	0.00759	0.01466	0.00304	19.19	33.88	123.8	1150	0.1181	0.1551	0.1459	0.09975	0.2948	0.08452	0		
13	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	54.16	0.00577	0.04061	0.02791	0.01282	0.02008	0.00414	20.42	27.28	136.5	1299	0.1396	0.5609	0.3965	0.181	0.3792	0.1048	0		
14	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.00314	0.08297	0.0889	0.0409	0.04484	0.01284	20.96	29.94	151.7	1332	0.1037	0.3903	0.3639	0.1767	0.3176	0.1023	0		
15	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.00977	0.03126	0.05051	0.01992	0.02981	0.003	16.84	27.66	112	876.5	0.1131	0.1924	0.2322	0.1119	0.2809	0.06287	0		
16	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.00643	0.05936	0.05501	0.01628	0.01961	0.00809	15.03	32.01	108.8	697.7	0.1651	0.7725	0.6943	0.2208	0.3596	0.1431	0		
17	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.00561	0.0424	0.04741	0.0109	0.01857	0.00547	17.46	37.13	124.1	943.2	0.1678	0.6577	0.7026	0.1712	0.4218	0.1341	0		
18	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.00572	0.01162	0.01998	0.01109	0.0141	0.00209	19.07	30.88	123.4	1138	0.1464	0.1871	0.2914	0.1609	0.3029	0.08216	0		
19	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.00703	0.02501	0.03188	0.01297	0.01689	0.00414	20.96	31.48	136.8	1315	0.1789	0.4233	0.4784	0.2073	0.3706	0.1142	0		
20	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.00649	0.01893	0.03391	0.01521	0.01356	0.002	27.32	30.88	186.8	2398	0.1512	0.315	0.5372	0.2388	0.2768	0.07615	0		
21	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.00846	0.0146	0.02387	0.01315	0.0198	0.0023	15.11	19.26	99.7	711.2	0.144	0.1773	0.239	0.1288	0.2977	0.07259	1		
22	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67	0.0041	0.01898	0.01698	0.00649	0.01678	0.00243	14.5	20.49	96.09	630.5	0.1312	0.2776	0.189	0.07283	0.3184	0.08183	1		
23	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.00961	0.01432	0.01985	0.01421	0.02027	0.00297	10.23	15.66	65.13	314.9	0.1324	0.1148	0.08867	0.06227	0.245	0.07773	1		
24	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384	44.91	0.00679	0.05328	0.06446	0.02252	0.03672	0.00439	18.07	19.08	125.1	980.9	0.139	0.5954	0.6305	0.2393	0.4667	0.09946	0		
25	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303	93.99	0.00473	0.01259	0.01715	0.01038	0.01083	0.00199	29.17	35.59	188	2615	0.1401	0.26	0.3155	0.2009	0.2822	0.07526	0		
26	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.0633	0.8068	0.9017	5.455	102.6	0.00605	0.01882	0.02741	0.0113	0.01468	0.0028	26.46	31.56	177	2215	0.1805	0.3578	0.4695	0.2095	0.3613	0.09564	0		
27	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	0.07413	1.046	0.976	7.276	111.4	0.00803	0.03799	0.03732	0.02397	0.02308	0.00744	22.25	21.4	152.4	1461	0.1545	0.3949	0.3853	0.255	0.4066	0.1059	0		
28	14.58	21.53	97.41	644.8	0.1054	0.1868	0.1425	0.08783	0.2252	0.06924	0.2545	0.9832	2.11	21.05	0.00445	0.03055	0.02681	0.01352	0.01454	0.00371	17.62	33.21	122.4	896.9	0.1525	0.6643	0.5539	0.2701	0.4264	0.1275	0		
29	18.61	20.25	122.1	1094	0.0944	0.1066	0.149	0.07731	0.1697	0.05699	0.8529	1.849	5.632	93.54	0.01075	0.02722	0.05081	0.01911	0.02293	0.00422	21.31	27.26	139.9	1403	0.1338	0.2117	0.3446	0.149	0.2341	0.07421	0		
30	15.3	25.27	102.4	732.4	0.1082	0.1697	0.1683	0.08751	0.1926	0.0654	0.439	1.012	3.498	43.5	0.00523	0.03057	0.03576	0.01083	0.01768	0.00297	20.27	36.71	149.3	1269	0.1641	0.611	0.6335	0.2024	0.4027	0.09876	0		
31	17.57	15.05	115	955.1	0.09847	0.1157	0.09875	0.07953	0.1739	0.06149	0.6003	0.8225	4.655	61.1	0.00563	0.03033	0.03407	0.01354	0.01925	0.00374	20.01	19.52	134.9	1227	0.1255	0.2812	0.2489	0.1456	0.2756	0.07919	0		
32	18.63	25.11	124.8	1088	0.1064	0.1887	0.2319	0.1244	0.2183	0.06197	0.8307	1.466	5.574	105	0.00625	0.03374	0.05196	0.01158	0.02007	0.00456	23.15	34.01	160.5	1670	0.1491	0.4257	0.6133	0.1848	0.3444	0.09782	0		
33	11.84	18.7	77.93	440.6	0.1109	0.1516	0.1218	0.05182	0.2301	0.07799	0.4825	1.03	3.475	41	0.00555	0.03414	0.04205	0.01044	0.02273	0.00567	16.82	28.12	119.4	888.7	0.1637	0.5775	0.6956	0.1546	0.4761	0.1402	0		
34	17.02	23.98	112.8	899.3	0.1197	0.1496	0.2417	0.1203	0.2248	0.06382	0.6009	1.398	3.999	67.78	0.00827	0.03082	0.05042	0.01112	0.02102	0.00385	20.88	32.09	136.1	1344	0.1634	0.3559	0.5588	0.1847	0.353	0.08482	0		
35	19.27	26.47	127.9	1162	0.09401	0.1719	0.1657	0.07593	0.1853	0.06261	0.5558	0.6062	3.528	68.17	0.00502	0.03318	0.03497	0.00964	0.01543	0.0039	24.15	30.9	161.4	1813	0.1509	0.659	0.6091	0.1785	0.3672	0.1123	0		
36	16.13	17.88	107	807.2	0.104	0.1559	0.1354	0.07752	0.1998	0.06515	0.334	0.6857	2.183	35.03	0.00419	0.02868	0.02664	0.00907	0.01703	0.00382	20.21	27.26	132.7	1261									

2. 딥러닝 모델 구현 실습

텐서플로를 사용한 딥러닝 모델 만들고 평가 지표를 알려줘.



텐서플로를 사용하여 간단한 딥러닝 모델을 만들어 보겠습니다. 이 모델은 주어진 데이터셋을 사용하여 유방암의 양성(benign) 또는 악성(malignant)을 예측하는 분류 모델입니다.

1. 데이터 준비 및 전처리
2. 모델 생성
3. 모델 학습
4. 모델 평가

이 과정을 통해 모델을 만들고 평가 지표를 확인해 보겠습니다.

1. 데이터 준비 및 전처리

먼저 데이터셋을 텐서플로에서 사용할 수 있도록 준비합니다.

```
python 코드 복사  
  
import pandas as pd  
import tensorflow as tf  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler  
  
# 데이터 로드  
file_path = '/mnt/data/Breast Cancer.csv'  
data = pd.read_csv(file_path)
```

**ChatGPT 현재 버전에서는
딥러닝 모델을 실행 시키지
못하므로 구글 코랩으로
소스를 복사하여 실행시킨다**

2. 딥러닝 모델 구현 실습

iris.csv 데이터 파일: 다중 분류 모델

	A	B	C	D	E
1	sepal length	sepal width	petal length	petal width	label
2	5.1	3.5	1.4	0.2	0
3	4.9	3	1.4	0.2	0
4	4.7	3.2	1.3	0.2	0
5	4.6	3.1	1.5	0.2	0
6	5	3.6	1.4	0.2	0
7	5.4	3.9	1.7	0.4	0
8	4.6	3.4	1.4	0.3	0
9	5	3.4	1.5	0.2	0
10	4.4	2.9	1.4	0.2	0
11	4.9	3.1	1.5	0.1	0
12	5.4	3.7	1.5	0.2	0
13	4.8	3.4	1.6	0.2	0
14	4.8	3	1.4	0.1	0
15	4.3	3	1.1	0.1	0
16	5.8	4	1.2	0.2	0
17	5.7	4.4	1.5	0.4	0
18	5.4	3.9	1.3	0.4	0
19	5.1	3.5	1.4	0.3	0
20	5.7	3.8	1.7	0.3	0

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows x 5 columns

2. 딥러닝 모델 구현 실습

텐서플로우를 사용한 딥러닝 모델 만들고 평가 지표를 알려줘.



아이리스(Iris) 데이터셋을 사용하여 딥러닝 모델을 만들고, 이를 평가해보겠습니다. 텐서플로우를 사용하여 모델을 생성, 학습, 평가하는 전 과정을 진행하겠습니다.

1. 데이터 준비 및 전처리

먼저 데이터셋을 텐서플로우에서 사용할 수 있도록 준비합니다.

python

코드 복사

```
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.utils import to_categorical

# 데이터 로드
file_path_iris = '/mnt/data/iris.csv'
data = pd.read_csv(file_path_iris)
```

**ChatGPT 현재 버전에서는
딥러닝 모델을 실행 시키지
못하므로 구글 코랩으로
소스를 복사하여 실행시킨다**

감사합니다