

Тақырыбы: Нейрондық Желілердің Негіздері және Терең Оқытудың Қазіргі Үлгілері

Нейрондық желілер (Neural Networks) — бұл адам миының биологиялық нейрондарына ұқсас, өзара байланысқан есептеуіш элементтердің (жасанды нейрондардың) жиынтығынан тұратын модельдер. Желінің негізгі функциясы — кіріс деректерден күрделі үлгілерді үйрену арқылы шешім қабылдау немесе болжам жасау.

Әрбір жасанды нейрон немесе перцептрон келесі математикалық операцияларды орындайды:

1. Кірістер және Салмақталған Қосынды: Нейрон бірнеше кірістерді  $x_1, x_2, x_n$  қабылдайды. Әрбір кіріс нейронның есептеуге қосқан үлесін көрсететін сәйкес салмаққа ( $w_1, w_2, w_n$ ) көбейтіледі.
2. Ығысудың Қосылуы (Bias): Салмақталған кірістердің қосындысына  $b$  ығысу (bias) мәні қосылады. Ығысу нейронға тіпті кірістер нөлге тең болса да, белсендірілуге мүмкіндік береді.

$$z = \left( \sum_{i=1}^n x_i w_i \right) + b$$

Мұндағы  $z$  — нейронның таза кірісі (net input).

3. Активация Функциясы (Activation Function): Таза кіріс  $z$  міндетті түрде активация функциясынан ( $f$ ) өтеді. Бұл функция нейронның соңғы шығысын  $a$  анықтайды және ең маңыздысы – желіге сызықтық емес (non-linear) қасиет береді, бұл күрделі, сызықтық емес мәселелерді шешу үшін қажет.

$$a = f(z)$$

| Активация Функциясы          | Формуласы                                          | Қолданылуы                                  |
|------------------------------|----------------------------------------------------|---------------------------------------------|
| Sigmoid (Сигмоид)            | $f(z) = \frac{1}{1+e^{-z}}$                        | Шығыс қабатында (Бинарлық классификация)    |
| ReLU (Rectified Linear Unit) | $f(z) = \max(0, z)$                                | Жасырын қабаттарда (Ең жиі қолданылатыны)   |
| Softmax (Софтмакс)           | $f(\mathbf{z})_k = \frac{e^{z_k}}{\sum_j e^{z_j}}$ | Шығыс қабатында (Көп сыныпты классификация) |

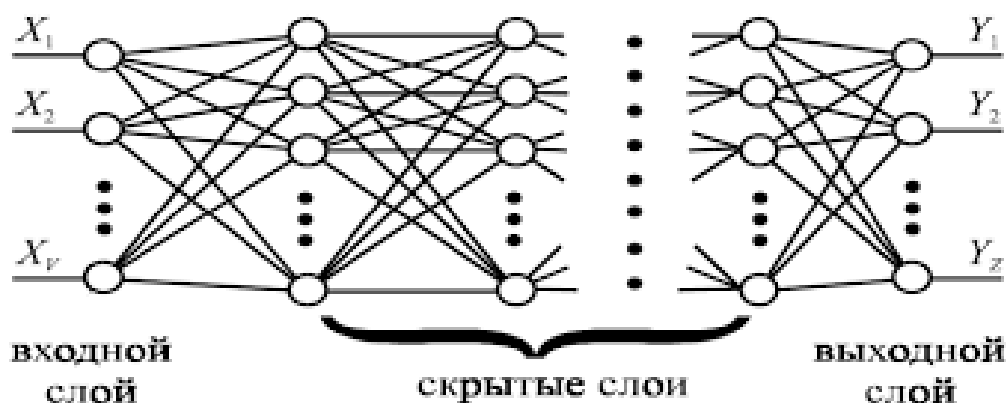
## Нейрондық Желінің Архитектурасы

Нейрондық желілер нейрондардың қабаттасып орналасуымен сипатталады:

- Кіріс қабаты (Input Layer): Модельге бастапқы деректерді қабылдайды. Нейрондар саны ерекшеліктер (фичуралар) санына тең.
- Жасырын қабаттар (Hidden Layers): Кіріс пен шығыс арасында орналасады. Бұл қабаттар деректердің күрделі өзара байланыстарын және иерархиялық ерекшеліктерін үйренеді. Жасырын қабаттағы нейрон саны арқылы желінің "ені" (width) анықталады.
- Шығыс қабаты (Output Layer): Желінің соңғы нәтижесін (болжамды) шығарады. Нейрондар саны шешілетін мәселеге (классификация, регрессия) байланысты.

Терең оқыту (Deep Learning) — бұл кем дегенде екі немесе одан да көп жасырын қабаты бар нейрондық желілерді (MLP, CNN, RNN, Transformer) оқытуды білдіреді. Қабаттардың көптігі желіге абстрактілі және күрделі концепцияларды үйренуге мүмкіндік береді.

Сурет 10.1: Көпқабатты Перцептрон (MLP) архитектурасы



### Құрылымдары мен оқыту әдістері

Оқыту процесі: Алға және Кері Тарату

Нейрондық желілерді оқыту (салмақтарды оңтайландыру) екі негізгі фазадан тұрады:

1. Алға Бағыттау (Forward Propagation):

Кіріс деректері кіріс қабатынан бастап, барлық жасырын қабаттар арқылы, активация функцияларымен өңделіп, шығыс қабатында соңғы болжамдық мән  $\hat{y}$  береді.

2. Шығын Функциясын Есептеу (Loss Calculation):

Модельдің болжамы  $\hat{y}$  мен деректегі нақты мән  $y$  арасындағы айырмашылық (қателік) шығын функциясы (Loss Function) арқылы есептеледі.

- Орташа Квадраттық Қате (MSE) Регрессия үшін

$$Loss = \frac{1}{m} \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2$$

- Кросс-Энтропия Классификация үшін:

$$L = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$$

### 3. Кері Тарату (Backpropagation):

Бұл алгоритм есептелген қателікті шығыс қабатынан бастап, желінің барлық жасырын қабаттарына дейін кері бағытта таратады. Негізгі мақсат — әрбір салмаққа қатысты шығын функциясының градиентін есептеу. Бұл градиент салмақтың шығынға қосқан үлесін анықтайды.

### 4. Салмақтарды Жаңарту (Оңтайландыру):

Градиенттік мәліметтер Градиенттік Түсу (Gradient Descent) әдісі арқылы салмақтарды жаңарту үшін қолданылады. Бұл процесс шығын функциясының ең төменгі нүктесіне (глобалдық минимумға) жетуге тырысады.

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial \text{Loss}}{\partial w}$$

Мұндағы  $\eta$  (Оқыту Жылдамдығы - Learning Rate) — салмақтардың бір қадамда қаншалықты жылдам жаңаратынын анықтайтын маңызды гиперпараметр.

Оңтайландыру және Артық Оқытудан Қорғау

- Оңтайландырушылар (Optimizers): SGD (Stochastic Gradient Descent), Adam, RMSprop сияқты жетілдірілген алгоритмдер градиенттік түсудің жылдамдығы мен тиімділігін арттырады.
- Төмендету (Regularization): Желінің оқыту деректеріндегі шуды жаттап алуын (артық оқыту) болдырмау үшін L2-төмендету немесе Dropout (кейбір нейрондарды уақытша өшіру) әдістері қолданылады.