

پاسخ سوال جلسه دوم

سوال : توابع مورد استفاده در شبکه عصبی مصنوعی را با فرمول و کاربرد بیان کنید.

تابع فعالساز (Activation Functions): در شبکه‌های عصبی برای ایجاد عدم خطیت و تعمیم‌پذیری استفاده می‌شوند. این توابع معمولاً به عنوان تابعی غیر خطی در هنگام اعمال ورودی‌ها به لایه‌های شبکه استفاده می‌شوند، تا شبکه قادر به یادگیری روابط پیچیده‌تر و استخراج ویژگی‌های غیر خطی از داده‌ها باشد. در زیر، توابع فعال‌سازی رایج و کاربردهای آنها توضیح می‌دهم:

۱- تابع سیگموید (Sigmoid): زمانی که می‌خواهیم خروجی مدل احتمال باشد، از تابع سیگموید استفاده می‌کنیم؛ چون تابع سیگموید مقادیر را به بازه صفر تا ۱ می‌برد و احتمالات هم میان همین بازه قرار دارند. این تابع در مسائل طبقه‌بندی معمولاً خیلی خوب عمل می‌کند.

معادله $f(x) = s = 1/(1+e^{-x})$ بازه: (0,1)

۲- تابع تانژانت هایپربولیک (Tanh, Hyperbolic Tangent): این تابع صفر محور است؛ مقادیر منفی، به شدت منفی و مقادیر صفر در گراف تانژانت هایپربولیک نزدیک به صفر نگاشت می‌شوند.

معادله $f(x) = a = \tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ بازه: (-1, 1)

۳- تابع واحد یک‌سوسدهی خطی (ReLU / Rectified Linear Unit): تابع فعالساز واحد یک‌سوسدهی خطی در زمینه‌ی یادگیری عمیق بسیار مشهور است و در بیشتر مواقع استفاده می‌شود. این تابع به این صورت عمل می‌کند که مقادیر منفی (زیر صفر) را صفر و مقادیر مثبت (بیشتر از صفر) و مقادیر برابر با صفر را همان مقدار خودش در نظر می‌گیرد. از نظر محاسباتی بسیار کارآمد است و به شبکه اجازه می‌دهد به سرعت همگرا شود؛ زیرا رابطه‌ی آن خطی است و به همین دلیل، در مقایسه با تابع‌های سیگموید و Tanh، سریع‌تر است.

معادله $f(x) = a = \max(0, x)$ بازه: (0, +∞)

۴- تابع فعالساز Leaky ReLU: این تابع فعالساز برای حل مشکل اصلی تابع ReLU ارائه شده است از مشکل مرگ ReLU جلوگیری می‌کند. این تابع یک شیب مثبت ملایم به سمت مقادیر منفی دارد که این امر باعث می‌شود عملیات پس انتشار (Backpropagation) حتی برای مقادیر منفی هم انجام شود.

معادله $f(x) = a = \max(0.01x, x)$ بازه: (0.01, +∞)

۵- تابع (Softmax): این تابع فعالساز در طبقه‌بندی‌های چندکلاسه استفاده می‌شود. زمانی که احتیاج داشته باشیم در خروجی احتمال عضویت بیشتر دو کلاس را پیش‌بینی کنیم، می‌توانیم به‌سراغ این تابع برویم. تابع سافت‌مکس تمامی مقادیر یک بردار با طول K را به بازه‌ی صفر تا ۱ می‌برد، به‌طوری که جمع تمامی مقادیر این بردار با هم ۱ می‌شود. این تابع برای نورون‌های لایه‌ی خروجی استفاده می‌شود؛ زیرا در شبکه‌های عصبی در آخرین لایه (خروجی) به طبقه‌بندی ورودی‌ها در کلاس‌های مختلف نیاز داریم. این تابع قابلیت استفاده در تسک‌های چندکلاسه را دارد. خروجی هر کلاس را میان صفر تا ۱ عادی‌سازی می‌کند؛ سپس آن‌ها را بر مجموعه‌شان تقسیم و احتمال عضویت مقادیر ورودی را در هر کلاس به ما در خروجی ارائه می‌کند.

$$\text{معادله } (f(x) = e^{x_i} / (\sum_{i=0}^K e^{x_i})) \text{ بازه: } (0,1)$$

۶- تابع هزینه (Cost Function): استفاده برای اندازه‌گیری خطا یا تفاوت بین خروجی تخمینی شبکه و خروجی مورد انتظار، و به عنوان معیاری برای تنظیم و بهینه‌سازی وزن‌ها در فرآیند آموزش شبکه. مانند: میانگین مربعات خطا (MSE):

$$\text{معادله } MSE = \frac{1}{n} \sum (\hat{Y} - Y)^2$$

تابع بهینه‌سازی (Optimization Function): در شبکه‌های عصبی مصنوعی، برای بهینه‌سازی وزن‌ها و پارامترهای دیگر شبکه از توابع بهینه‌سازی استفاده می‌شود. در زیر، چند نمونه از توابع بهینه‌سازی نام برده و کاربرد و فرمول هرکدام را توضیح داده شده است:

۷- تابع Gradient Descent: این تابع بهینه‌سازی برای به دست آوردن وزن‌های بهینه بر اساس مقدار گرادیان (شیب) تابع هدف استفاده می‌شود. به طور معمول، از گرادیان نسبت به وزن‌ها استفاده می‌شود تا به سمت جهت کاهش تابع هدف حرکت کند.

$$\text{معادله: وزن جدید} = \text{وزن قبلی} - \text{نرخ یادگیری} \times \text{گرادیان تابع هدف نسبت به وزن}$$

۸- تابع Stochastic Gradient Descent: این تابع بهینه‌سازی نیز برای به دست آوردن وزن‌های بهینه استفاده می‌شود، اما در آن از نمونه‌های تصادفی داده‌های آموزش برای محاسبه گرادیان استفاده می‌شود. این روش به طور کلی سرعت آموزش را افزایش می‌دهد.

$$\text{معادله: وزن جدید} = \text{وزن قبلی} - \text{نرخ یادگیری} \times \text{گرادیان تابع هدف نسبت به وزن (برای هر نمونه)}$$

$$\text{آموزشی به‌طور تصادفی)}$$

۹- تابع (Adam (Adaptive Moment Estimation : این تابع بهینه‌سازی یک ترکیبی از روش‌های

Momentum و RMSprop است. از آن برای بهبود عملکرد بهینه‌سازی و سرعت آموزش استفاده می‌شود.

معادله : وزن جدید = وزن قبلی - نرخ یادگیری × مقدار اصلاح‌شده گرادیان

۱۰- تابع (Adagrad (Adaptive Gradient : این تابع بهینه‌سازی برای سفرهای آموزشی با شیب نسبتاً بزرگ

استفاده می‌شود. آن را می‌توان برای مسائلی که از طریق تحلیل گرادیان مستقیم انجام می‌شوند به کار برد.

معادله : وزن جدید = وزن قبلی - نرخ یادگیری × گرادیان تقسیم بر جمع مربعات گرادیان‌های قبلی