

অধ্যায় ১১ : পারসেপট্রন (Perceptron)

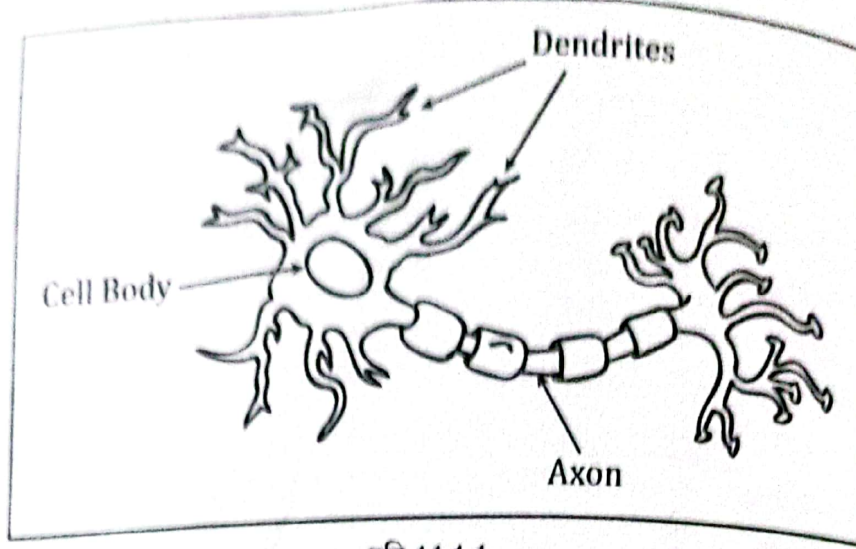
পারসেপট্রন (Perceptron) – নাম শুনেই মনে হচ্ছে বেশ ভারি কিছু একটা, তাই না? আসলেই, এটিকে বেশ রাজকীয় অ্যালগরিদমই বলা চলে। পারসেপট্রন (সাধারণত পারসেপট্রন বলতে Single Layer পারসেপট্রনকে বোঝায়) একটি বাইনারি ক্লাসিফায়ার অ্যালগরিদম। বাইনারি ক্লাসিফায়ার মানে হচ্ছে, এটি নির্ধারণ করতে পারে যে, কোনো বস্তু কোনো একটি নির্দিষ্ট ক্লাসের কি না। যদি দুইয়ের অধিক ক্লাস থাকে, সে ক্ষেত্রে এই সিংগেল লেয়ার পারসেপট্রন ব্যবহার করা হয় না। এটি এক ধরনের লিনিয়ার ক্লাসিফায়ার এবং সুপারভাইজড লার্নিংয়ের মাধ্যমে ব্যবহৃত হয়ে থাকে।

এই পারসেপট্রন প্রথম উদ্ভাবন করা হয় 1957 সালে কর্নেল অ্যারোনটিক্যাল ল্যাবরেটরি (Cornell Aeronautical Laboratory)-তে। এর উদ্ভাবক ছিলেন ফ্র্যাংক রোসেনব্ল্যাট (Frank Rosenblatt, 1928-1971)। একটি খুব মজার খবর বলি, 1957 সালে যখন এই পারসেপট্রন প্রথম ব্যবহৃত হয়, তখন থেকেই ধীরে ধীরে কম্পিউটার ও মেশিন লার্নিংয়ের ভবিষ্যৎ, দিয়ে উঠতে থাকে। সেসবের ওপরে ভিত্তি করেই, সেই সময়ে 'দ্য নিউ ইয়র্ক টাইমস' (The New York Times) পত্রিকায় পারসেপট্রন সম্পর্কে লেখা হয় – 'the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.'

পরবর্তী সময়ে ধীরে ধীরে এই সিংগেল লেয়ার পারসেপট্রন থেকেই মাল্টি লেয়ার পারসেপট্রন বা নিউরাল নেটওয়ার্কের জন্ম হয়, যা নিয়ে আজকের এই পৃথিবীতে এত মাতামাতি। পারসেপট্রন নিয়ে পড়তে গেলে প্রথমে আমাদের জানতে হবে নিউরন (Neuron) কী এবং এর আদলে মিল রেখে কীভাবে একটি গাণিতিক মডেল তৈরি করা যায়। সেখান থেকে পরবর্তী সময়ে পারসেপট্রন সম্পর্কে ধারণা দেওয়া হবে।

পরিচ্ছেদ ১১.১ : নিউরন (Neuron) এবং এর আদলে গাণিতিক কাঠামো

আমরা স্কুল-কলেজে অনেকেই জীববিজ্ঞানে নিউরন সম্পর্কে জেনেছি। নিউরন হচ্ছে আমাদের মস্তিষ্কের গাঠনিক একক, এক ধরনের কোষ। নিউরন দেখতে কীরকম সে সম্পর্কে মোটামুটি একটি ধারণা পাওয়া যাবে নিচের ছবিটি থেকে (ছবি 11.1.1) :



ছবি 11.1.1

একটি নিউরনের মোট তিনটি প্রধান অংশ থাকে, এর কোষদেহ (Cell Body), ডেনড্রাইট (Dendrite) নামের ছোটো ছোটো শাখা-প্রশাখা, যা কোষদেহ থেকে বের হয়েছে এবং থাকে একটি লম্বা দণ্ডের মতো অংশ – অ্যাক্সন (Axon)। এই অ্যাক্সনগুলো আবার পরবর্তী নিউরনের ডেনড্রাইটের সঙ্গে আষ্টেপৃষ্ঠে সংযুক্ত থাকে, এই সংযোগকে বলা হয় সিন্যাপস (Synapse)। এই সিন্যাপস সংযোগগুলোর কারণেই আমরা চিন্তা করতে পারি, স্মৃতি ধারণ করতে পারি, সিদ্ধান্ত নিতে পারি। নিউরনের ক্ষেত্রে ডেনড্রাইটগুলোকে বলা যেতে পারি ইনপুট ঢোকার পথ, আর অ্যাক্সনকে বলতে পারি আউটপুট বের হবার পথ।

এখন, আমাদের মেশিন লার্নিং পড়ার সময় এই জীববিজ্ঞানে পড়ে আসা নিউরনের কাঠামোর সঙ্গে সামঞ্জস্যপূর্ণ একটি গাণিতিক কাঠামো সম্পর্কে পড়তে হবে, এর নামও নিউরন, শুধু পার্থক্য এই যে এটি গণিতে ব্যবহার করা হয় এবং এর ইনপুট, আউটপুটগুলো হয় বিভিন্ন সংখ্যা।

আমরা এবারে গাণিতিক নিউরনের একটি ছবি দেখে ফেলি (ছবি 11.1.2) :

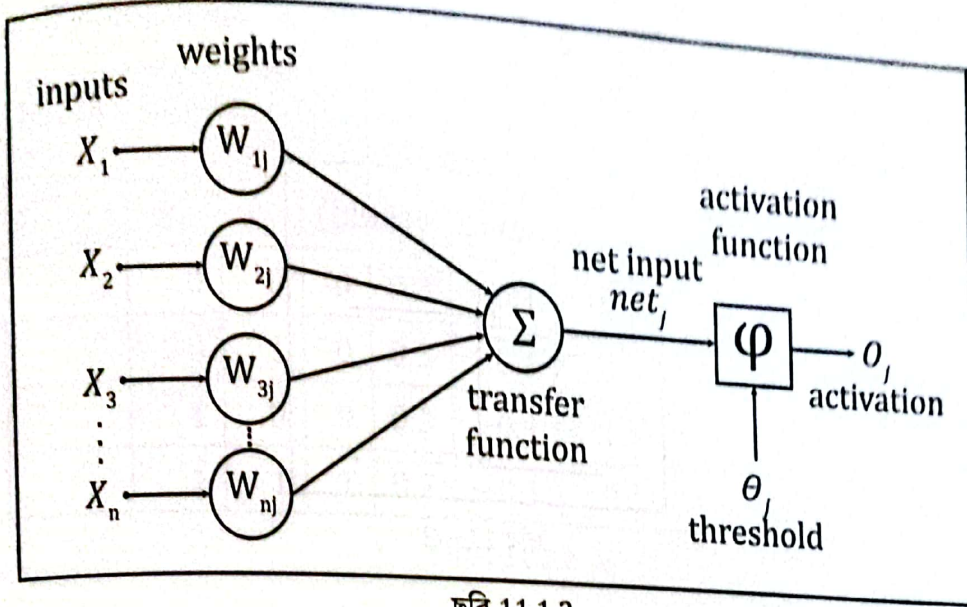
এখানে $x_1, x_2 \dots x_n$ ইত্যাদি হলো একটি ট্রেনিং ডেটা পয়েন্টের বিভিন্ন ফিচারের মান। এখানে ডেটার ডাইমেনশন n ।

J দিয়ে J -তম ডেটা পয়েন্ট বোঝানো হচ্ছে। আর w হচ্ছে স্বভাবতই আমাদের প্রতিটি ডেটা পয়েন্টের সঙ্গে সম্পৃক্ত ওয়েইট-এর মান।

সহজ ভাষায় এখন বলে দিই পারসেপট্রন কীভাবে কাজ করবে –

- প্রথমে প্রতিটি ইনপুটের সঙ্গে তার ওয়েইট গুণ হয়ে সবগুলো একসঙ্গে যোগ করা হবে। এই কাজটি করবে ট্রান্সফার ফাংশন (Transfer Function) নামে একটি ফাংশন, যেটি দেখানো হয়েছে Σ চিহ্ন দিয়ে। এটি একটি অ্যাডার জাংশন (Adder Junction) বা সামিং জাংশন

(Summing Junction)। এর কাজ হচ্ছে, যা ইনপুট পাবে সব যোগ করবে। এর ভেতরে ইনপুট হিসেবে দেওয়া হবে $x_1w_1, x_2w_2 \dots x_nw_n$ ইত্যাদি। বরাবরের মতোই আমরা এই ওয়েইটগুলোতে প্রথমে র্যান্ডম মান বসাব। এই সামিং জংশন সবগুলোকে যোগ করে আউটপুট দেবে $x_1w_1 + x_2w_2 + \dots + x_nw_n$ । একেই আমরা নিট ইনপুট (net input) net_j হিসেবে চিহ্নিত করেছি।



ছবি 11.1.2

- পরবর্তী সময়ে এই net_j -কে একটি থ্রেশহোল্ড (Threshold) মানের সঙ্গে তুলনা করা হবে। এই তুলনা করার কাজটি করবে অ্যাক্টিভেশন ফাংশন (Activation Function) নামে একটি ফাংশন। এই থ্রেশহোল্ড মানকে আমরা আপাতত T_h দিয়ে চিহ্নিত করতে পারি। এখানে উল্লেখ্য যে, আমাদের পারসেপট্রন কিন্তু শুধুই একটি লিনিয়ার ক্লাসিফায়ার ছাড়া আর কিছুই নয়। যদিও সাধারণত, অ্যাক্টিভেশন ফাংশনগুলো সাধারণত কোনো লিনিয়ার ফাংশনকে নন-লিনিয়ারিটি দেয়, কিন্তু, পারসেপট্রনের ক্ষেত্রে সেটি হয় না (এর কারণ হিসেবে বলা যায়, পারসেপট্রনের ইনপুট এবং আউটপুট ছাড়া মধ্যবর্তী আর কোনো গোপন বা হিডেন (hidden) হিডেন লেয়ার নেই, এ ব্যাপারটি পরবর্তী অধ্যায়ে আলোচনা করা হয়েছে)।
- যদি, net_j অর্থাৎ $\sum_{i=1}^n x_i w_i \geq T_h$ হয়, তাহলে আমাদের পারসেপট্রন আউটপুট দেবে 1, নাহলে আউটপুট দেবে 0। এ ধরনের ফাংশনকে ইউনিট স্টেপ ফাংশন (Unit Step Function)-ও বলা হয়।

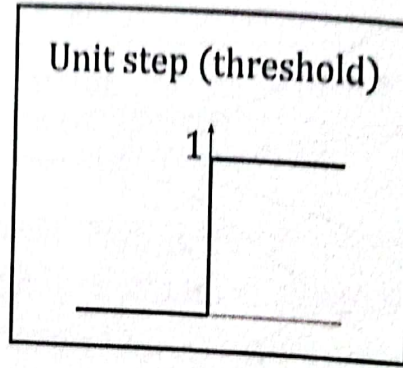
ছবি 11.1.3-তে একটি ইউনিট স্টেপ ফাংশন দেখানো হয়েছে।

- ধরা যাক, পারসেপট্রন আউটপুট দিল Y_p আর ডেটা পয়েন্ট X-এর জন্য প্রকৃত আউটপুট হওয়ার কথা Y। এখন, যদি $Y = Y_p$ হয়, তার মানে কোনো এরর হয়নি, সুতরাং আমাদের আর ওয়েইটের মান আপডেট করা লাগবে না। কিন্তু যদি $Y \neq Y_p$ না হয়, তখন আমাদের ওয়েইটের মানগুলো আপডেট করতে হবে।

ওয়েইটের মান আপডেট করার সূত্র হলো -

$$W_l(j+1) = W_l(j) + \alpha \cdot x_l \cdot (Y - Y_p)$$

এখানে, $W_l(j+1)$ মানে হচ্ছে নতুন ওয়েইট, $W_l(j)$ হচ্ছে পুরোনো ওয়েইট, α হচ্ছে আমাদের সেই লার্নিং রেট, যা আমরা গ্রেডিয়েন্ট ডিসেন্ট অ্যালগরিদমে পড়েছিলাম, x_l হচ্ছে ওয়েইট $W_l(j)$ যে ইনপুটের সঙ্গে সম্পর্কযুক্ত (অর্থাৎ, l -তম ট্রেনিং ডেটা) এবং $(Y - Y_p)$ হচ্ছে আমাদের এরর।



ছবি 11.1.3

পরিচ্ছেদ ১১.২ : পারসেপট্রন ট্রেনিং দেওয়া

এখন আমরা সরাসরি কীভাবে একটি পারসেপট্রনকে ট্রেনিং দিতে হয় সেটি দেখব। আমাদের এই কাজের জন্য ডেটাসেটটি নিম্নরূপ :

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

টেবিল 11.2.1

ওপরের টেবিল দেখে অনেকেই হয়তো বুঝে গেছেন, এটি দুটি ইনপুটের জন্য লজিক্যাল অ্যান্ড (Logical AND) অপারেশনের ট্রুথ টেবিল (Truth Table)। এটিই হবে আমাদের ইনপুট, আমরা চাই আমাদের এই পারসেপট্রন (0,0) ইনপুট পেলে 0 আউটপুট দেবে; (1,1) ইনপুট পেলে 1 আউটপুট দেবে ইত্যাদি।

অধ্যায় ১১ : পারসেপট্রন (Perceptron)

এখন তাহলে আমাদের ট্রেনিং শুরু করা যাক।

আমরা আমাদের এই উদাহরণের জন্য ধরে নিচ্ছি আমাদের প্রাথমিক ওয়েইট হচ্ছে 0.3 এবং -0.1। তবে কেউ চাইলে অন্য কোনো মানও ধরে নিতে পারেন, শূন্য ধরে নেওয়াটা সবচেয়ে ভালো বুদ্ধি। এই উদাহরণের জন্য $\alpha = 0.1, T_h = 0.2$ ।

আমাদের ট্রেনিং কোনো রাউন্ডে (epoch) সব এররের মান শূন্য না হওয়া পর্যন্ত চলতে থাকবে।

Epoch	Inputs		Y	Initial Weights		Y_p	Error ($Y - Y_p$)	Final Weights	
	x_1	x_2		w_1	w_2			w_1	w_2
1	0	0	0	0.3	-0.1	0	0	0.3	-0.1
	0	1	0	0.3	-0.1	0	0	0.3	-0.1
	1	0	0	0.3	-0.1	1	-1	0.2	-0.1
	1	1	1	0.2	-0.1	0	1	0.3	0.0

টেবিল 11.2.2

প্রথম রাউন্ডে দেখুন, (1,0) ইনপুটের জন্য আউটপুট হওয়ার কথা 0, কিন্তু আউটপুট হলো 1,

সুতরাং, $error = -1$

আউটপুট 1 হওয়ার কারণ,

$$x_1 w_1 + x_2 w_2 = 1 \times 0.3 + 0 \times (-0.1) = 0.2$$

আমরা বলেছিলাম, যদি $x_1 w_1 + x_2 w_2 \geq T_h$ হয়, তাহলে আমাদের পারসেপট্রন আউটপুট দেবে 1, আর নাহলে আউটপুট দেবে 0। যেহেতু আমাদের $T_h = 0.2$ ছিল এবং তা $x_1 w_1 + x_2 w_2$ রাশিটির সমান, তাই পারসেপট্রনের আউটপুট হলো 1।

প্রথম ওয়েইট আপডেট হয়ে দাঁড়াল, $0.3 + [0.1 \times 1 \times (-1)] = 0.2$ এবং
দ্বিতীয় ওয়েইট আপডেট হয়ে দাঁড়াল, $-0.1 + [0.1 \times 0 \times (-1)] = -0.1$

এভাবেই বাকি হিসাবনিকাশ করতে হবে পরের রাউন্ডগুলোর জন্য। এই রাউন্ডেই হিসাব থেমে যাবে না, কেননা আমাদের পারসেপট্রন এই রাউন্ডে একবার ভুল করেছে। যতক্ষণ পর্যন্ত না এমন কোনো রাউন্ড পাওয়া যাবে, যেখানে এই পারসেপট্রন কোনো এরর দেবে না, ততক্ষণ পর্যন্ত আমাদের ট্রেনিং চালিয়ে যেতে হবে।

পরের রাউন্ডের হিসাবনিকাশগুলো দেখে নেওয়া যাক,

দ্বিতীয় রাউন্ড :

Epoch	Inputs		Y	Initial Weights		Y_p	Error ($Y - Y_p$)	Final Weights	
	x_1	x_2		w_1	w_2			w_1	w_2
2	0	0	0	0.3	0.0	0	0	0.3	0.0
	0	1	0	0.3	0.0	0	0	0.3	0.0
	1	0	0	0.3	0.0	1	-1	0.2	0.0
	1	1	1	0.2	0.0	1	0	0.2	0.0

টেবিল 11.2.3

এই রাউন্ডেও দেখুন একবার ভুল আছে, সুতরাং আবারও আমাদের ট্রেনিং দিতে হবে। এখানে উল্লেখ্য, এই রাউন্ডের প্রাথমিক ওয়েইট 0.3 এবং 0.0 আমরা পেয়েছি আগের রাউন্ডের সর্বশেষ ফাইনাল ওয়েইট থেকে। আরেকটি ব্যপার, এপক (Epoch) মানে সহজ ভাষায় রাউন্ড নম্বর বোঝাচ্ছে।

তৃতীয়, চতুর্থ ও পঞ্চম রাউন্ড :

Epoch	Inputs		Y	Initial Weights		Y_p	Error ($Y - Y_p$)	Final Weights	
	x_1	x_2		w_1	w_2			w_1	w_2
3	0	0	0	0.2	0.0	0	0	0.2	0.0
	0	1	0	0.2	0.0	0	0	0.2	0.0
	1	0	0	0.2	0.0	1	-1	0.1	0.0
	1	1	1	0.1	0.0	0	1	0.2	0.1
Epoch	Inputs		Y	Initial Weights		Y_p	Error ($Y - Y_p$)	Final Weights	
	x_1	x_2		w_1	w_2			w_1	w_2
4	0	0	0	0.2	0.1	0	0	0.2	0.1
	0	1	0	0.2	0.1	0	0	0.2	0.1
	1	0	0	0.2	0.1	1	-1	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1

অধ্যায় ১১ : পারসেপট্রন (Perceptron)

Epoch	Inputs		Y	Initial Weights		Y_p	Error ($Y - Y_p$)	Final Weights	
	x_1	x_2		w_1	w_2			w_1	w_2
5	0	0	0	0.1	0.1	0	0	0.1	0.1
	0	1	0	0.1	0.1	0	0	0.1	0.1
	1	0	0	0.1	0.1	0	0	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1

টেবিল 11.2.3

যাক, অবশেষে এই পঞ্চম রাউন্ডে এসে আমরা দেখলাম যে আমাদের পারসেপট্রন আর কোনো ভুল করেনি। তাই, আমাদের পারসেপট্রনকে ট্রেনিং দেওয়া এই রাউন্ডেই সমাপ্ত হবে। চাইলে একে আরো কয়েক রাউন্ড ট্রেনিং দেওয়া যায়, কিন্তু তাতে আমাদের কস্ট (cost) খুব একটা কমবে না।

এখানে কিন্তু আমরা স্টোকাস্টিক গ্রেডিয়েন্ট ডিসেন্ট (Stochastic Gradient Descent) পদ্ধতি প্রয়োগ করেছি, বুঝতেই পারছেন। গ্রেডিয়েন্ট ডিসেন্ট কীভাবে কাজ করে, এটি যাঁরা ইতিমধ্যেই কিছুটা ভুলে গেছেন, তাঁরা পরিচ্ছেদ ৩.৮ আরেকবার পড়ে নিতে পারেন।

এই ছিল কীভাবে একটি পারসেপট্রনকে ট্রেনিং দিতে হয় সেই সংক্রান্ত আলোচনা। আশা করি, সবাই পারসেপট্রনের ধারণা বুঝতে পেরেছেন এবং নিজে নিজেই এখন কোড করে ফেলতে পারবেন।