

ডিপ লার্নিং ও ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং

রবিউল আউয়াল

২৫ ডিসেম্বর, ২০২১



# সূচীপত্র

0.1	পরিচিতি . . . . .	5
I	পার্ট ১ . . . . .	7
1	ম্যাশিন লার্নিং . . . . .	9
1.1	সুপারভাইজড লার্নিং সেটআপ . . . . .	9
1.1.1	ইনপুট, আউটপুট . . . . .	9
1.1.2	মডেল . . . . .	9
1.1.3	কস্ট ফাংশন . . . . .	9
2	ম্যাশিন লার্নিং এলগরিদমস . . . . .	11
2.0.1	লজিস্টিক রিগ্রেশন . . . . .	11
2.0.2	লিনিয়ার রিগ্রেশন . . . . .	11
2.0.3	রেগুলারাইজেশন . . . . .	11
3	ডীপ লার্নিং . . . . .	13
3.1	নিউরাল নেটস . . . . .	13
3.2	ট্রেনিং নিউরাল নেটস . . . . .	13
3.2.1	ব্যাকপ্রোপাগেশন . . . . .	13
3.2.2	অপটিমাইজেশন . . . . .	13
3.3	প্র্যাক্টিক্যাল টিপস ও পাইটর্চ . . . . .	13
II	পার্ট ২ . . . . .	15
4	টার্মিনোলজিঃ হাতি ঘোড়া বাঘ . . . . .	17
4.0.1	ল্যাপসুয়েজ . . . . .	17
4.0.2	সিকুয়েন্স . . . . .	17
4.0.3	সিমান্টিকস . . . . .	17
4.0.4	সিনট্যাক্স . . . . .	17
4.0.5	কনটেক্সট . . . . .	17
4.0.6	ফাংশন এপ্রোক্সিমেশন . . . . .	17
4.0.7	ল্যাংগুয়েজ এজ ফাংশন এপ্রোক্সিমেশন . . . . .	17
5	ডিস্ট্রিবিউটেড সিমান্টিকস . . . . .	19
5.1	ডিস্ট্রিবিউটেড ওয়ার্ড রিপ্রেজেন্টেশন . . . . .	19
5.2	নিউরাল ওয়ার্ড এম্বেডিংস . . . . .	19
5.2.1	ওয়ার্ড ভেকটরস (word2vec) . . . . .	19
5.2.2	স্কিপ গ্রাম (Skip-gram) . . . . .	19
5.2.3	কন্টিনিউয়াস ব্যাগ অব ওয়ার্ডস (CBOW) . . . . .	19
5.2.4	নেগ্যাটিভ স্যামপ্লিং . . . . .	20
5.2.5	নিউরাল ওয়ার্ড এম্বেডিংস এভালুয়েশন . . . . .	20
5.2.6	আউট অব ভোকাবুলারি ওয়ার্ডস ও ক্যারেক্টার কমপোজিশনাল মডেল . . . . .	20
5.3	কন্টেক্সচুয়াল এম্বেডিংস . . . . .	20
5.3.1	Cove . . . . .	20

5.3.2	ELMO . . . . .	20
6	সিকুয়েন্স ল্যাবেলিং . . . . .	21
6.1	সিকুয়েন্স ল্যাবেলিং ও ক্লাসিফিকেশন . . . . .	21
6.2	সিকুয়েন্স ল্যাবেলিং এপ্লিকেশন . . . . .	21
6.2.1	পার্টস অফ স্পিচ ট্যাগিং . . . . .	21
6.2.2	নেইমড এন্টিটি রিকগনিশন . . . . .	21
7	লিঙ্গুইস্টিক স্ট্রাকচার . . . . .	23
7.0.1	ডিপেন্ডেন্সি পার্সার . . . . .	23
7.0.2	ইউনিভার্সাল ডিপেন্ডেন্সিস . . . . .	23
7.1	রেফারেন্স রেজুলেশন . . . . .	23
III	পার্ট ৩ . . . . .	25
8	রিকারেন্ট নিউরাল নেটওয়ার্কস ও ল্যাপুয়েজ মডেল . . . . .	27
8.0.1	n গ্রাম ল্যাপুয়েজ মডেল . . . . .	27
8.0.2	রিকারেন্ট নিউরাল নেটস . . . . .	27
8.0.3	ট্রেনিং রিকারেন্ট নিউরাল নেটস . . . . .	27
8.0.4	এক্সপ্লোডিং ও ভ্যানিশিং গ্র্যাডিয়েন্ট . . . . .	29
8.0.5	লং শর্ট টার্ম মেমোরি নেটওয়ার্ক . . . . .	29
8.0.6	দুটি অত্যাবশ্যকীয় উপাদানঃ নিউরাল এম্বেডিংস + রিকারেন্ট ল্যাপুয়েজ মডেল . . . . .	29
9	ম্যাশিন ট্রান্সলেশন ও সিকুয়েন্স টু সিকুয়েন্স মডেল . . . . .	31
9.0.1	ম্যাশিন ট্রান্সলেশন টাস্ক . . . . .	31
9.0.2	স্ট্যাটিস্টিক্যাল ম্যাশিন ট্রান্সলেশন . . . . .	31
9.0.3	সিকুয়েন্স টু সিকুয়েন্স মডেল . . . . .	31
10	এটেনশন মেকানিজম . . . . .	33
10.0.1	বাহদান এটেনশন . . . . .	33
10.0.2	মাল্টি হেড এটেনশন . . . . .	34
10.0.3	সেলফ এটেনশন . . . . .	34
10.0.4	সেক২সেক এপ্লিকেশন . . . . .	34
10.0.5	ট্রান্সফর্মার . . . . .	34
10.0.6	সেলফ এটেনশন ও কনভল্যুশনের মধ্যকার সম্পর্ক . . . . .	34
11	রিকার্ডিভ ও কনভল্যুশনাল নিউরাল নেটস ফর ল্যাপুয়েজ প্রসেসিং . . . . .	35
11.0.1	রিকার্ডিভ নিউরাল নেটস . . . . .	35
11.0.2	কনভল্যুশন . . . . .	35
11.0.3	কনভল্যুশন ফর সেনটেন্স মডেলিং . . . . .	35
11.0.4	কনভল্যুশন ফর সেনটেন্স ক্লাসিফিকেশন . . . . .	36
11.0.5	কনভল্যুশন ট্রেনিংঃ কিছু প্রাক্টিক্যাল ট্রিকস . . . . .	36
11.0.6	রিকারেন্ট ও কনভল্যুশনাল নিউরাল নেটওয়ার্কের তুলনামূলক আলোচনা . . . . .	36
12	প্রিট্রেন্ড ল্যাপুয়েজ মডেলস . . . . .	37
12.0.1	কনটেক্সটুয়াল রাপ্রেজেন্টেশন . . . . .	37
12.0.2	প্রি-ট্রেনিং . . . . .	37
12.0.3	বার্ট (BERT) . . . . .	37
12.0.4	জিপিটি ৩ (GPT 3) . . . . .	37
IV	পার্ট ৪ . . . . .	39
13	GLUE বেসমার্ক টাস্কস . . . . .	41
13.1	সেন্টেন্স ক্লাসিফিকেশন . . . . .	41
13.2	ন্যাচারাল ল্যাপুয়েজ ইনফারেন্স . . . . .	41
13.3	কোশেন্স আনসারিং . . . . .	41

13.3.1	রিডিং কম্প্রিহেনশন . . . . .	43
13.3.2	ওপেন ডোমেইন কুশ্চেন আনসারিং . . . . .	43
14	ট্রান্সফার লার্নিং ফর ল্যাঙ্গুয়েজ প্রসেসিং . . . . .	45
14.0.1	সিকুয়েন্সিয়াল ট্রান্সফার . . . . .	46
14.0.2	মাল্টি-টাস্ক ট্রান্সফার লার্নিং . . . . .	46
14.0.3	লাইট ওয়েট ফাইন-টিউনিং . . . . .	46
15	এথিকস ও ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং . . . . .	49
15.0.1	জেন্ডার বায়াস ও ওয়ার্ড এস্কেডিং . . . . .	49
15.0.2	ডিবায়াসিং . . . . .	49
15.0.3	স্টকাস্টিক প্যারটস . . . . .	49
16	ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং রিসার্চ . . . . .	51
16.0.1	থিম ১ঃ ল্যাঙ্গুয়েজ মডেলিং . . . . .	51
16.0.2	থিম ২ঃ কমনসেন্স রিজনিং . . . . .	51
16.0.3	থিম ২ঃ মডেল এনালাইসিস ও এক্সপ্লোরেশন . . . . .	51
16.0.4	থিম ৩ঃ মাল্টিলিঙ্গুয়াল ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং . . . . .	51
16.0.5	থিম ৪ঃ কন্টিনিউয়াল লার্নিং . . . . .	51

## 0.1 পরিচিতি

বাংলায় ন্যাচারাল ল্যাঙ্গুয়েজ বই লিখার চেষ্টা করছি। ২০২০ সালের মাঝামাঝি সময়ে আমরা একটা রিডিং গ্রুপ শুরু করি। রিডিং কোর্সের জন্য এই বইটি সহায়ক হিসেবে কাজ শুরু করি। আমাদের লেকচারগুলি থেকে গুরুত্বপূর্ণ কম্পোনেন্ট নিয়ে বইটি সাজানো হয়েছে। এতে প্রায় ১৬ টি অধ্যায় রয়েছে। মোট ৪ খণ্ডে বইটি ভাগ করা হয়েছে। এই বইকে ডিপ লার্নিং ফর ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং হিসেবে ব্যবহার করা যেতে পারে। ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং, বাংলায় বলছি ভাষাপ্রযুক্তি। যে টপিকগুলি নিয়ে আলোচনা করা হয়েছে সেগুলির মধ্যে থাকছে - প্রশ্ন উত্তর, ম্যাশিন ট্রান্সলেশন (অনুবাদ), চ্যাটবট, সিকুয়েন্স ল্যাবেলিং এগুলি হলো জনপ্রিয় সমস্যা। আমার নিজের গবেষণায় খুব একটা উন্নতির আশা দেখছি না। যেহেতু গত দুই বছর ধরে ভাষা প্রযুক্তি নিয়ে পড়ছি। আমি ভাবলাম এই ফিল্ডে যেসকল চমৎকার কাজ হয়েছে, সেগুলি নিয়ে আলাপ করতে পারলে কিছুটা মনোবাসনা পূর্ণ হবে। একটা কথা প্রচলিত আছে যেসব লোকেরা নাকি কবিতা লিখতে গিয়ে ব্যর্থ হইছেন তারাই নাকি পরবর্তীতে বাংলা বিভাগের অধ্যক্ষ হন এবং কবিদের ঘাড় মটকানোর কাজটি করে বেড়ান পেশাদারিত্বের সাথে। যদিও আমার ইচ্ছে সেরকম কিছু নয়। যাহোক মুগ্ধতার বয়ান আপনাদের সামনে উপস্থাপন করা যাক তবে।



খন্ড I

পার্ট ১





## অধ্যায় 1

# ম্যাশিন লার্নিং

### 1.1 সুপারভাইজড লার্নিং সেটআপ

#### 1.1.1 ইনপুট, আউটপুট

#### 1.1.2 মডেল

#### 1.1.3 কস্ট ফাংশন



## অধ্যায় 2

# ম্যাশিন লার্নিং এলগরিদমস

2.0.1 লজিস্টিক রিগ্রেশন

2.0.2 লিনিয়ার রিগ্রেশন

2.0.3 রেগুলারাইজেশন



## অধ্যায় 3

# ডীপ লার্নিং

3.1 নিউরাল নেটস

3.2 ট্রেনিং নিউরাল নেটস

3.2.1 ব্যাকপ্রোপাগেশন

3.2.2 অপটিমাইজেশন

3.3 প্রায়িক্যাল টিপস ও পাইটর্চ



খন্ড II

পার্ট ২





## অধ্যায় 4

# টার্মিনোলজিঃ হাতি ঘোড়া বাঘ

4.0.1 ল্যাঙ্গুয়েজ

4.0.2 সিকুয়েন্স

4.0.3 সিমান্টিকস

4.0.4 সিনট্যাক্স

4.0.5 কনটেক্সট

4.0.6 ফাংশন এপ্রোপ্রিমেশন

4.0.7 ল্যাংগুয়েজ এজ ফাংশন এপ্রোপ্রিমেশন



## অধ্যায় 5

# ডিস্ট্রিবিউটেড সিমান্টিকস

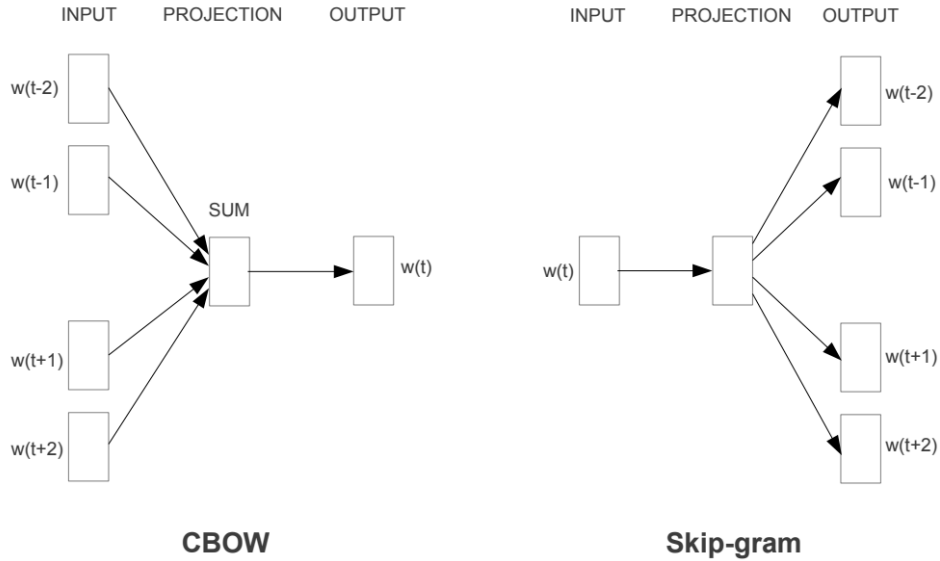
### 5.1 ডিস্ট্রিবিউটেড ওয়ার্ড রিপ্রেজেন্টেশন

আমাদের কাছে একটি বড় করপাস আছে। একটা বাক্যের টার্গেট এবং কনটেক্সট এই দুটো টার্ম আমাদের প্রথমে বুঝতে হবে। বাক্যগুলিতে ম্যাশিনের ইনপুট হিসেবে ব্যবহার করার জন্য আমাদের নিউমারিক্যাল ভ্যালুতে রূপান্তর করতে হবে। ওয়ান-হট-ভেক্টর। যেখানে একটা ভেক্টরের সবগুলি আইটেমের মান শূন্য শুধু একটা আইটেমের মান ১। ওয়ার্ড টু ভেক্টর হলো একটা টেবিল। একে ওয়ার্ড এম্বেডিং বলা হয়। যেকোন বাক্যের এম্বেডিং নিউরাল নেটওয়ার্কের জন্য ইনপুট বা ফিচার হিসেবে কাজ করে। ওয়ার্ড ভেক্টর অনেক সিমান্টিক তথ্য সংগ্রহ করে রাখতে পারে যা নিউরাল নেটওয়ার্ক ট্রেনিংয়ে সাহায্য করে। কন্টিনিউয়াস ব্যাগ অফ ওয়ার্ডস মডেল (সিভোও)। আরেকটি মডেল হচ্ছে স্কিপ গ্রাম মডেল। আমরা যেকোন একটি মডেল ব্যবহার করতে পারি।

### 5.2 নিউরাল ওয়ার্ড এম্বেডিংস

#### 5.2.1 ওয়ার্ড ভেকটরস (word2vec)

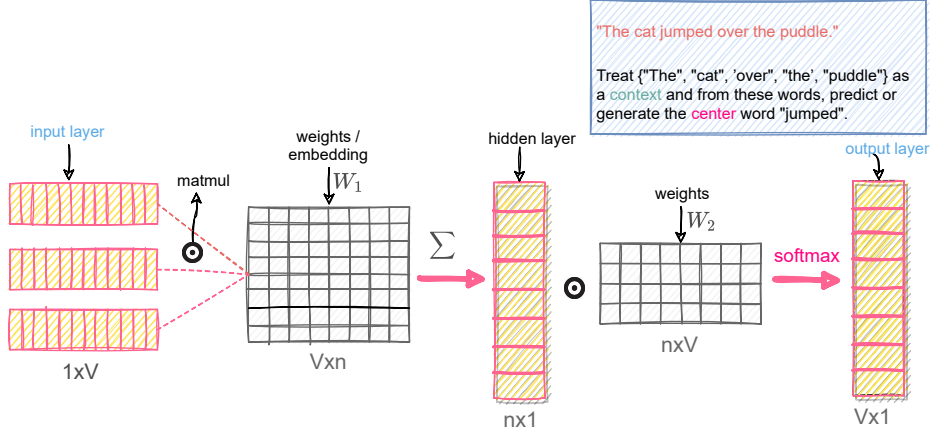
ওয়ার্ড টু ভেক হলো আমরা একটা টেবিল চিন্তা করতে পারি। সেই টেবিল থেকে ওয়ার্ড যেখানে ম্যাচ করছে সেই ভ্যালুটা রিটার্ন করবে।



চিত্র 5.1: ওয়ার্ড টু ভেক মডেল

#### 5.2.2 স্কিপ গ্রাম (Skip-gram)

#### 5.2.3 কন্টিনিউয়াস ব্যাগ অব ওয়ার্ডস (CBOW)



চিত্র 5.2: কন্টিনিউয়াস ব্যাগ অব ওয়ার্ডস মডেল

### 5.2.4 নেগ্যাটিভ স্যামপ্লিং

নেগেটিভ স্যামপ্লিং হলো পজিটিভ উদাহরণের সাথে নেগেটিভ উদাহরণ ব্যবহার করা। একে কন্ট্রাসটিভ লার্নিং ও বলে।

### 5.2.5 নিউরাল ওয়ার্ড এম্বেডিংস এভালুয়েশন

### 5.2.6 আউট অব ভোকাবুলারি ওয়ার্ডস ও ক্যারেক্টার কমপোজিশনাল মডেল

## 5.3 কন্টেক্সুয়াল এম্বেডিংস

### 5.3.1 Cove

:

### 5.3.2 ELMO

: আমরা ২০১৮ সালে। এখনো সবাই লং শর্ট টার্ম মেমোরি নেটওয়ার্ক ব্যবহার করছে। আমরা দেখেছি কোভে বাইডিরেকশনাল এলএসটিএম ব্যবহার করেছে, ২ লেয়ার। এলমো এসে বলে আমরা আরো গভীর যাবো। কোভ এলএসটিএমের শেষ লেয়ার থেকে হিডেন স্টেট নেয়। এলমো প্রতিটা এলএসটিএম থেকে হিডেন স্টেট নেয় এবং সবগুলি মার্জ করে ফেলে। এখানেও আমাদের ওয়ার্ড এম্বেডিংস লাগবে। আমাদের কাছে একটা বড় করপাস থাকবে। সেটার বাক্য আমরা ওয়ার্ডটুকেকে দিব। তারপর সেগুলি ইনপুট হিসেবে পরের লেয়ারে দিব। আগের লেয়ারের আউটপুট আবার পরের লেয়ারে ইনপুট হবে। এভাবে আমরা স্ট্যাক করবো। গ্লোভ হলো নন-কন্টেক্সুয়াল এম্বেডিংস। এলমো হচ্ছে কন্টেক্সুয়াল। প্রতিটা এলএসটিএমের হিডেন স্টেট জোড়া দিবো। আমরা বাইডিরেকশনাল এলএসটিএম ব্যবহার করবো। একটা ফরোয়ার্ড ল্যাংগুয়েজ মডেল এবং একটা ব্যাকওয়ার্ড ল্যাংগুয়েজ মডেল। ল্যাংগুয়েজ মডেল ব্যবহার করার কারণেই এটা কন্টেক্সুয়াল। কারণ পরের টোকেন প্রেডিক্ট করার জন্য আমরা আগের টোকেনগুলিকে কন্টেক্সট হিসেবে ব্যবহার করবো।

$$\sum_{k=1}^N (\log p(t_k | t_1, \dots, t_{k-1}; \phi_x, \theta_{LSTM}, \theta_s) + \log p(t_k | t_k + 1, \dots, t_N; \phi_x, \theta_{LSTM}, \theta_s)) \quad (5.1)$$

যেখানে  $\theta_x$  ইনপুট,  $\theta_{LSTM}$  এলএসটিএম আর  $\theta_s$  হচ্ছে সফটম্যাক্স প্যারামিটার এবং  $N$  হচ্ছে বাক্যের মধ্যে কতগুলি টোকেন আছে।  $\log p()$  হচ্ছে ম্যাক্সিমাম লাইকলিহুড।

## অধ্যায় ৬

# সিকুয়েন্স ল্যাবেলিং

### 6.1 সিকুয়েন্স ল্যাবেলিং ও ক্লাসিফিকেশন

### 6.2 সিকুয়েন্স ল্যাবেলিং এপ্লিকেশন

#### 6.2.1 পার্টস অফ স্পিচ ট্যাগিং

#### 6.2.2 নেইমড এন্টিটি রিকগনিশন

ল্যাংগুয়েজ প্রোসেসিংয়ে আমরা যে নেটওয়ার্কগুলি ব্যবহার করে থাকি সেগুলি যেকোন একটা ব্যবহার করতে পারি। আমরা ওয়ার্ড এমবেডিং ইউজ করতে পারি, এলএসটিএম ব্যবহার করতে পারি বা কনভলিউশন ব্যবহার করতে পারি। সবগুলি নেটওয়ার্কের শেষে আমরা একটা লিনিয়ার লেয়ার ব্যবহার করবো। লিনিয়ার লেয়ারের পরে একটা সফটম্যাক্স প্রেডিকশন হবে। তখন সে আমাদের যতগুলি ট্যাগ আছে সেগুলির আউটপুট দিবে। যদি ৯ টা ক্লাস থাকে সেগুলির থেকে যেকোন আউটপুট হবে। ল্যাংগুয়েজ ইনপুট সিকুয়েন্স চলক দৈর্ঘ্যের হয়ে থাকে। কোন বাক্যের দৈর্ঘ্য হতে পারে ১০, আরেকটি ১২, আরেকটি ৩০। আমরা বাক্যগুলিকে রিশেইপ করে একটি ফিক্সড লেংথে নিয়ে আসবো। রিকারেন্ট নেটওয়ার্ক একটা ফিক্সড লেংথ ইনপুট চায়। আমার বাক্যের দৈর্ঘ্য যদি ১০ হয়, তাহলে বাকি পজিশনগুলিতে আমি ০ দিয়ে দিবো। ও থাকা মানে এই জায়গাগুলিতে অতিরিক্ত প্যাডিং করা হয়েছে। প্যাডিং মানে বাকি ২০ টা টোকেন অতিরিক্ত dummy ও দিয়ে ভরে ফেলা। এখন সিকুয়েন্স প্রেডিকশনের কাজে এই প্যাডিং কাজে দিবে। আমি যে টোকেনগুলি পাচ্ছি, তার যেগুলির মান শূন্য হবে সেগুলির জন্য আমি লস হিসেব করবো না। অই স্থানে মডেলের আউটপুটের জন্য কোন লস কম্পিউট করা হবে না। কারণ যেহেতু ভ্যালু ও তাই এটা আমরা প্যাড করে এনেছি। আপনি যদি মাক্সিং ব্যবহার করেন তাহলে ১০ টা সিকুয়েন্সের জন্য ১০ টা আউটপুট পাবেন এবং লস ১০ টা টোকেনের জন্য আসবে। নিজের কোডিং করতে গেলে এই বিষয়টি মাথায় রাখতে হবে। আপনার নেটওয়ার্ক ফিক্সড লেংথ কিন্তু ইনপুট ভ্যারিয়েবল লেংথ। নেটওয়ার্কগুলি নিজেরা স্মার্ট না তাই এটা এক্সলিসিটলি হ্যান্ডেল করতে হবে যে এখানে আউটপুট হবে না। প্রশ্নঃ যদি ভ্যালু ০ শূন্য হয় তাহলে যদি বাদ দিয়ে দিই, তাহলে মূল সিকুয়েন্সে ০ থাকে? এক্ষেত্রে সচেতন থাকতে হবে যে এম্বেডিং ০ থাকবে না কোন টোকেন এনকোড করার জন্য। যদি এমন হয় এম্বেডিং ০ আছে, তাহলে আমরা -১০০০ দিয়ে মাক্সিং করে নিতে পারি। Glove এমবেডিং ০ নেই, তাই আমরা ০ ব্যবহার করতে পারি। এবার তাহলে রিকারেন্ট নেটওয়ার্ক দিয়ে নেইমড এন্টিটি রিকগনিশন নিয়ে আলোচনা করা যাক।

আমরা যখন ইমপ্লিমেন্ট করবো তখন আউটপুট লেয়ার থাকবে। BIO ট্যাগ থাকে। B - শুরু ট্যাগ, I - ইন্টারমিডিয়েট ট্যাগ এবং O - আউটপুট ট্যাগ। এখন আমরা যদি লং শর্ট টার্ম মেমোরি নেটওয়ার্ক ব্যবহার করি তাহলে সমস্যা দেখা দিতে পারি। মাথায় রাখতে হবে আমরা একটি স্ট্রাকচার প্রেডিকশন টাস্ক নিয়ে কাজ করছি। যদি কোন পার্সন শুরু হয় তাহলে পার্সন ট্যাগ ইন্টারমিডিয়েট আসবে বা শেষ হবে। পার্সন শুরু হবার পর অর্গানাইজেশন শুরু বা এন্ড হতে পারবে না। এলএসটিএম কি ধরনের সমস্যা পড়তে পারে এ ধরনের স্ট্রাকচারড প্রেডিকশন হ্যান্ডেল করতে গেলে। দুটি শব্দ দিয়ে একই এন্টিটি রিপ্রেজেন্ট করে তখন কি হবে? আমরা স্ট্রাকচারড কোডিং স্কিম থেকে জানি কোন ট্যাগ শুরু হলে আগে সেই ট্যাগ শেষ হতে হবে। এলএসটিএম এর আগের স্টেপে আউটপুট প্রেডিকশন জানা নেই। আগের টোকেনের প্রেডিকশন জানা নেই, তাই ডিসিশনগুলি সবসময় লোকাল। এটার সমাধান কি হতে পারে? এর সমাধান হতে পারে আউটপুটের মধ্যে ডিপেন্ডেন্সি তৈরি করা। আগের স্টেপে যা আউটপুট জেনারেট করলাম সেটা পরের স্টেপকে জানিয়ে দেয়া। এখন প্রশ্ন হচ্ছে আমরা কি ধরনের নেটওয়ার্ক দিয়ে এই ডিপেন্ডেন্সি জেনারেট করতে পারি। হিডেন মার্কভ মডেল জানা থাকলে এই আলোচনা করা খুবই সহজ। মার্কভ প্রোপার্টি মনে করে যে আপনার যদি আগের স্টেটের তথ্য থাকে তাহলে পরের স্টেট অই স্টেটের উপর কন্ডিশন করলে বাকি স্টেটগুলি না জানলেও হবে। আমরা এখানে আবহাওয়ার তাপমাত্রা উদাহরণ হিসেবে দেখতে পারি। আমরা যদি গতকালের তাপমাত্রা উপর কন্ডিশন করি তাহলে এর আগে কিছু আমাদের এর আগের তাপমাত্রা জানার প্রয়োজন নেই। আগের ঘটনাগুলির সাপেক্ষে আজকের ঘটনাটি ইন্ডিপেন্ডেন্ট। কারণ হচ্ছে গতকালের ঘটনা আগের ঘটনাগুলির তথ্য ক্যারি করে। কন্ডিশনাল র্যান্ডম ফিল্ড লেয়ার ব্যবহার করে এই সমস্যার সমাধান সম্ভব।

CRF লেয়ার আউটপুট সিকুয়েন্স  $y$  এর প্রবাবিলিটি নির্ণয় করে নিচের ইকুয়েশন দিয়ে, যেখানে  $x$  ইনপুট সিকুয়েন্স।

$$P(y|x) = \frac{\exp \phi(x, y)}{\sum_{y'} \exp \phi(x, y')} = \frac{\exp \phi(x, y)}{Z(x)} \quad (6.1)$$

এখানে  $\phi(x, y)$  স্কোরিং ফাংশন, নিউরাল নেটওয়ার্ক। সম্ভাব্যতা হচ্ছে একটা বিশাল বড় সফটম্যাক্স। হরে যে  $\sum$  সেটা সকল সিকুয়েন্সের সম্ভাব্যতা হিশেব করে। এই বিশাল  $\sum$  কে বলা হয় পার্টিশন ফাংশন  $Z(x)$ .

## অধ্যায় 7

# লিঙ্গুইস্টিক স্ট্রাকচার

7.0.1 ডিপেন্ডেন্সি পার্সার

7.0.2 ইউনিভার্সাল ডিপেন্ডেন্সিস

7.1 রেফারেন্স রেজুলেশন





খন্ড III

পাট ৩



## অধ্যায় ৪

# রিকারেন্ট নিউরাল নেটওয়ার্কস ও ল্যাঙ্গুয়েজ মডেল

### ৪.০.১ n গ্রাম ল্যাঙ্গুয়েজ মডেল

ল্যাঙ্গুয়েজ মডেল কি

ল্যাঙ্গুয়েজ মডেল হচ্ছে পরবর্তী স্ট্রিং প্রেডিকশন টাস্ক। এটাকে ক্লাসিফিকেশন টাস্ক হিসেবে ভাবা যেতে পারে।

n গ্রামস

ল্যাঙ্গুয়েজ মডেল অভালুয়েশন

স্মুদিং

### ৪.০.২ রিকারেন্ট নিউরাল নেটস

রিকারেন্ট নিউরাল নেটওয়ার্কে [?] এক ধরনের চেইন থাকে। এটাকে ফিডব্যাক লুপ বলা হয়। চেইন বা সিকুয়েন্সিয়াল বিধায় এই ধরনের নেটওয়ার্ক সিকুয়েন্স নিয়ে ভালো কাজ করতে পারে। আরএনএন (RNN) এবং মাল্টিলেয়ার পার্সেপ্টন মূলত একই রকম, একমাত্র পার্থক্য হলো আরএনএন এর হিডেন ইউনিটসগুলির মধ্যে কানেকশন রয়েছে। চিন্তা করা যাক, একটা টেবিলের উপর অনেকগুলি নিউরন একটার একটা এক সারিতে বসানো হলো। এটা হলো আমাদের মাল্টিলেয়ার পার্সেপ্টন। এই নিউরনগুলির আগেরটির সাথে যদি পরেরটাকে জোড়া দেয়া হয় তবে সেটি হবে রিকারেন্ট নেটওয়ার্ক। সাধারণত প্রতি টাইম স্টেপে মডেল আগের টাইম স্টেপের হিডেন স্টেটকে ইনপুট হিসেবে নেয়। নেটওয়ার্কে ইনপুট  $x$ , হিডেন স্টেট  $h$  এবং আউটপুট  $y$ । রিকারেন্ট নেটওয়ার্কে টাইম স্টেপ  $t - 1$  থেকে  $t$  একটা কানেকশন থাকবে। কানেকশন থাকা বলতে বুঝাচ্ছি  $t - 1$  এর হিডেন স্টেট  $h_{t-1}$  পরের স্টেপ  $t$  এর ইনপুট।

$$h_t = F(h_{t-1}, x_t, ) \quad (8.1)$$

$$h_t = W_{rec}\sigma(h_{t-1}) + Ux_t + b \quad (8.2)$$

$$y_t = F(Vh_t) \quad (8.3)$$

নেটওয়ার্ক প্যারামিটার  $U, W$  ও  $V$  শেয়ারড প্রতি স্টেপে। প্যারামিটার  $W$  রিকারেন্ট। ফিডব্যাক লুপের এই প্রোপার্টি ন্যাচারাল ল্যাঙ্গুয়েজের জন্য বেশ কাজের। সাধারণত এইয়ে আমি এখন লিখছি একটা শব্দের পর আরেকটা শব্দের গাঁথুনি হচ্ছে। দেয়াল বানাতে যেমন একটা পর একটা ইটের গাঁথুনি দেয়। আমার আগের শব্দ যা লিখলাম তার উপর নির্ভর করে পরের শব্দটা লিখবো। এটা কিন্তু র্যান্ডম না। এখানে আগের শব্দের বা শব্দগুচ্ছের (prefix) একটা গুরুত্বপূর্ণ প্রভাব থাকবে। ম্যাশিন লার্নিং বিশেষ করে ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিংয়ে অধিকাংশ ডাটা সিকুয়েন্সিয়াল। টেক্সট, অডিও, ভিডিও ইত্যাদি সবকটা সিকুয়েন্সিয়াল। রিকারেন্ট নেট এই সিকুয়েন্স ইনফর্মেশনের মাধ্যমে ভালো কন্টেক্সট সংগ্রহ করতে পারে। এই সিকুয়েন্স অর্ডার যদি নেটওয়ার্কে আমরা ইনপুট হিসেবে না পাঠাতে পারি তাহলে মডেল ট্রেন করা সম্ভব হবে না অধিকাংশ ক্ষেত্রে। আশা করি কেন রিকারেন্ট নেটওয়ার্ক আর্কিটেকচার হিসেবে আমরা নির্বাচন করছি তার একটা ব্যাখ্যা দিতে পারলাম।

### ৪.০.৩ ট্রেনিং রিকারেন্ট নিউরাল নেটস

এবার আমরা নেটওয়ার্ক ট্রেনিং এর দিকে নজর দিব। এই নেটওয়ার্ক আমরা কিভাবে ট্রেন করতে পারি? এটা অন্যান্য নিউরাল নেট ট্রেনিং থেকে আলাদা কিছু নয়। এখানেও আমরা স্টোকাস্টিক গ্র্যাডিয়েন্ট ডিসচেন্ট (SGD) ব্যবহার করবো। রিকারেন্ট নিউরাল নেটের লস

হলো প্রতি স্টেপের লসের যোগফল।

$$E = \sum_{t=0}^T E_t \quad (8.4)$$

$$\frac{\partial E}{\partial U} = \sum_{t=0}^T \frac{\partial E_t}{\partial U} \quad (8.5)$$

ব্যাকপ্রোগেশন গ্রু টাইম (বিপিটিটি)

আমরা ব্যাকপ্রোগেশন ব্যবহার করবো নিউরাল নেটওয়ার্ক ট্রেন করার জন্য। বুঝার সুবিধার্থে আমরা এই লস ফাংশন নিয়ে বিস্তারিত চিন্তা করবো এখন। আমরা যদি একটা স্টেপ নিই,  $t$  থেকে  $t+1$  পজিশনে যাই। তাহলে আমাদের লস ফাংশনের  $E$  ডেরিভেটিভ ইনপুট  $U$  এর সাপেক্ষে দাঁড়ায়।

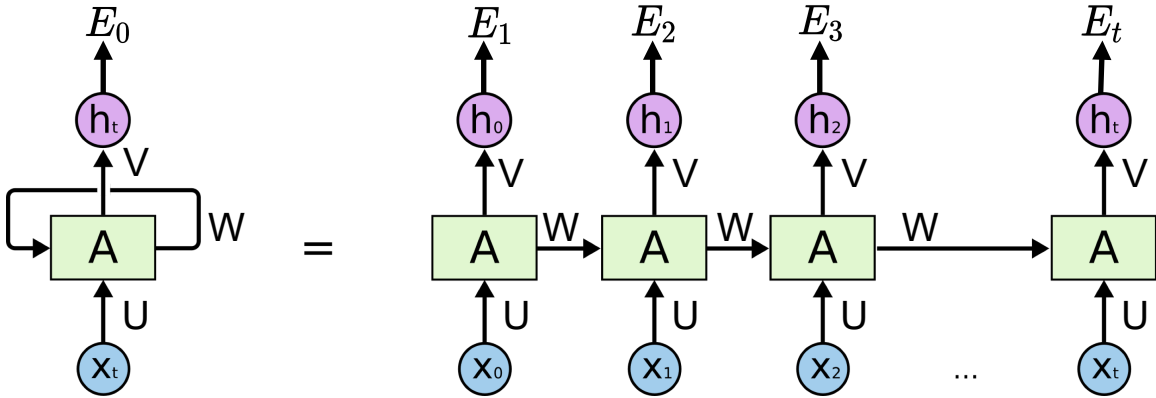
$$\frac{\partial E_{t+1}}{\partial U} = \frac{\partial E_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial U} = \frac{\partial E_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_t} \frac{\partial h_t}{\partial U} \quad (8.6)$$

$$(8.7)$$

ব্যাকপ্রোগেশন গ্রু টাইম হবে -

$$\frac{\partial E_{t+1}}{\partial U} = \sum_{k=0}^{t+1} \frac{\partial E_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k} \frac{\partial h_k}{\partial U} \quad (8.8)$$

$$(8.9)$$



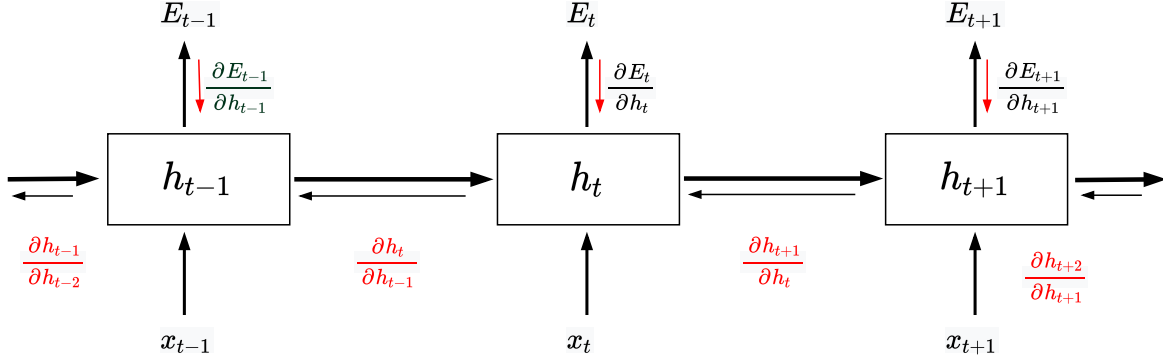
চিত্র ৪.১: রিকারেন্ট নেট ইরর

এখানে একটা ট্রিক আমরা ইউজ করবো যার সংক্ষিপ্ত নাম বিপিটিটি [?]. আমরা যদি লম্বা একটা সিকুয়েন্সের উপর ব্যাকপ্রোগেশন করতে যাই সেটা মেমোরি এক্সপেনসিভ এবং কম্পিউটেশনালি ক্লো হতে পারে। তাই আমরা একটা এপ্রোক্সিমেশন ব্যবহার করবো।

ট্রাকার ফোর্চিং

লং টার্ম ডিপেন্ডেন্সি

আমরা দেখতে পাচ্ছি রিকারেন্ট নেটের টাইম স্টেপ  $t$  তার আগের স্টেপ  $t-1$  এর উপর নির্ভরশীল। তাহলে আমাদের সিকুয়েন্স যদি ১০০ টা স্টেপ থাকে তবে সর্বশেষ ধাপ ১০০, তার আগের ধাপ ৯৯ এর উপর নির্ভরশীল। এখানে আমরা রিকারেন্ট ব্যাপারটা খেয়াল করতে পারি। কারণ ৯৯ তম ধাপ তার আগের ধাপ ৯৮ এর উপর নির্ভরশীল। এভাবে করে ৯৮ ধাপ ৯৭ এর উপর নির্ভরশীল এবং সবার শেষের ০ তে এসে থামবে। লক্ষ্য করা যাক, ১০০ তম ধাপ চেইন ধরে ৯৯ থেকে ০ সবগুলি ধাপের উপরও নির্ভর করছে। কারণ একটাকে কম্পিউট করতে গেলে তার আগেরটার ইনফরমেশন ছাড়া কম্পিউটেশন করা সম্ভব না। এই ধরনের ডিপেন্ডেন্সিকে বলা হয় লং টার্ম ডিপেন্ডেন্সি।



চিত্র ৪.২: রিকারেন্ট নেট ব্যাকপ্রোপাগেশন গ্রু টাইম

#### ৪.০.৪ এক্সপ্লোডিং ও ভ্যানিশিং গ্র্যাডিয়েন্ট

রিকারেন্ট নিউরাল নেট ভ্যানিশিং গ্র্যাডিয়েন্ট প্রব্লেম হয়। সহজ কথায় বললে, মডেল একটা লম্বা সিকুয়েন্স নিয়ে কাজ করতে গেলে আগে যেসব ইনপুট পেয়েছিল সেসব ভুলে যায়। ভ্যানিশিং গ্র্যাডিয়েন্টের জন্য আমাদেরকে জ্যাকোভিয়ান টার্মটা বুঝতে হবে।

জ্যাকোভিয়ান

গ্র্যাডিয়েন্ট ক্লিপিং

#### ৪.০.৫ লং শর্ট টার্ম মেমোরি নেটওয়ার্ক

আরএনএন এর ভ্যানিশিং গ্র্যাডিয়েন্ট সমস্যা সমাধান করার জন্য লং শর্ট টার্ম মেমোরি নেটওয়ার্ক ব্যবহার করা হয়।

মেমোরি সেল

যেকোন কম্পিউটেশনাল মেমোরিতে দুই ধরনের অপারেশন সম্ভবঃ ১) রিড ও ২) রাইট। যদি প্রথমবার লিখা হয় সেটাকে বলা হয় ইনসার্ট আর যদি পুরানো লিখা মুছে (ভুলে গিয়ে) নতুন কিছু লিখা হয় সেটাকে বলা হয় আপডেট।

ইনপুট, ফরগেট ও আউটপুট গেট

এলএসটিএম মেমোরি সেলে তথ্য আপডেট করার জন্য ৩ টি গেট ব্যবহার করা হয়। এখানে আমরা ধাপে ধাপে ৩ টি গেটের কাজ বুঝার চেষ্টা করবো।

$$f_t = \sigma(W_f \cdot [H_{t-1}, x_t] + b_f) \quad (8.10)$$

$$i_t = \sigma(W_i \cdot [H_{t-1}, x_t] + b_i) \quad (8.11)$$

$$\tilde{C}_t = \tanh(W_c \cdot [H_{t-1}, x_t] + b_c) \quad (8.12)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (8.13)$$

$$o_t = \sigma(W_o \cdot [H_{t-1}, x_t] + b_o) \quad (8.14)$$

$$h_t = o_t * \tanh(C_t) \quad (8.15)$$

$$(8.16)$$

#### ৪.০.৬ দুটি অত্যাৱশ্যকীয় উপাদানঃ নিউরাল এন্সেডিংস + রিকারেন্ট ল্যাঙ্গুয়েজ মডেল

আমরা যদি সিকুয়েন্সিয়াল ডাটা নিয়ে কাজ করতে চাই তাহলে আমাদের এখন দুটি উপাদান লাগবে- ১) ওয়ার্ড ভেক্টরস বা নিউরাল এমবেডিংস ও ২) রিকারেন্ট নিউরাল নেটওয়ার্ক। আর এই নেটওয়ার্ক ট্রেন করতে আমরা ব্যবহার করবো বিপিটিটি এলগরিদম। ওয়ার্ড ভেক্টরস আমাদের ফিচার আর রিকারেন্ট নেট আমাদের ডিপ লার্নিং টুল। সাধারণত আমরা এলএসটিএম (LSTM) ব্যবহার করার চেষ্টা

করবো কারণ ওর লং টার্ম ডিপেন্ডেন্সি নিয়ে কাজ করার সক্ষমতা বেশি। গ্র্যাডিয়েন্ট ইস্যুর কারণে রিকারেন্ট নেট প্র্যাক্টিক্যালি ট্রেন করা কষ্টসাধ্য কাজ। মোটাদাগে, আধুনিক ভাষা প্রযুক্তির সমস্যা সমাধানে এই দুই অস্ত্র থাকলেই কাজ চালিয়ে নেয়া যাবে!

কিছু টিপস# সিকুয়েন্সিয়াল ইনপুট আউটপুট বিষয়ক কিছু সূক্ষ্ম বিষয় আছে যা খেয়াল রাখা জরুরি। আমরা যখন কোড লিখবো বা মডেল পার্ফরমেন্স এনালাইসিস করবো তখন এই বিষয়গুলি কাজে দিবে। সিকুয়েন্স মডেলিংয়ের কাজ করতে গেলে আমাদের নিচের কয়েকটি বিষয় মাথায় রাখতে হবে।

- (i) ভ্যারিয়েবল লেংথ সিকুয়েন্স নিয়ে কাজ করতে হতে পারে (যেকোন টেক্সট ইনপুট; অথচ ছবি চাইলে আমরা ফিক্সড ডাইমেনশন রূপান্তর করে নিতে পারি)
- (ii) শব্দের অর্ডার মনে রাখা (ভাষার বাক্য গঠন বৈশিষ্ট্য)
- (iii) লং টার্ম ডিপেন্ডেন্সি মনে রাখা (সমৃদ্ধ কন্টেক্সট এর জন্য দরকার)
- (iv) সিকুয়েন্সগুলির মধ্যে প্যারামিটার (নেটওয়ার্ক ওয়েটস) শেয়ার করা (ট্রেনিং এফিশিয়েন্সির জন্য; প্রতি স্টেপে আলাদা প্যারামিটার থাকলে প্যারামিটারের সংখ্যা লিনিয়ারলি বেড়ে যাবে!)

## অধ্যায় ৯

# ম্যাশিন ট্রান্সলেশন ও সিকুয়েন্স টু সিকুয়েন্স মডেল

### ৯.০.১ ম্যাশিন ট্রান্সলেশন টাস্ক

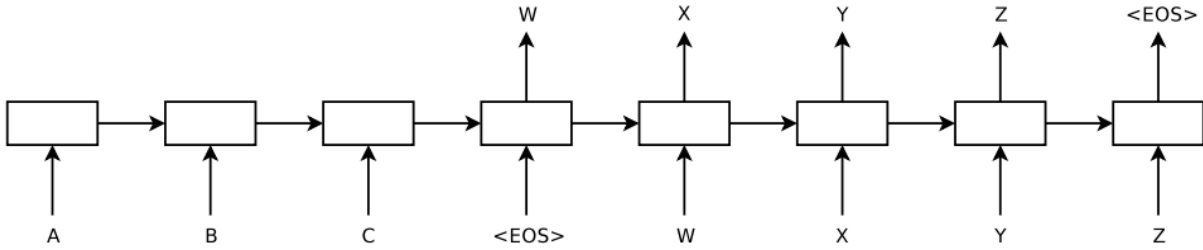
### ৯.০.২ স্ট্যাটিস্টিক্যাল ম্যাশিন ট্রান্সলেশন

স্ট্যাটিস্টিক্যাল ম্যাশিন ট্রান্সলেশনের ৩ টি কম্পোনেন্টঃ ট্রান্সলেশন মডেল, ল্যাঙ্গুয়েজ মডেল ও ডিকোডিং।

### ৯.০.৩ সিকুয়েন্স টু সিকুয়েন্স মডেল

এনকোডার ডিকোডার

আমরা যদি বাংলায় যদি একটা বাক্যকে অনুবাদ করতে চাই তাহলে আমাদের বাংলা বাক্যটি হবে ইনপুট আর ইংরেজি বাক্যটি পাবো আউটপুটে। ইনপুট বাক্যকে নিবে এনকোডার। আর এনকোডার এর ইনপুট থেকে আউটপুট বাক্য দিবে ডিকোডার। আমরা এনকোডার ডিকোডার দুটোর জন্য রিকারেন্ট নিউরাল নেট ব্যবহার করতে পারি। আমাদের একটা আরএনএন (RNN) লাগবে এনকোডারের জন্য আরেকটা লাগবে আউটপুট ডিকোড করতে।



চিত্র ৯.১: এনকোডার ডিকোডার মডেল



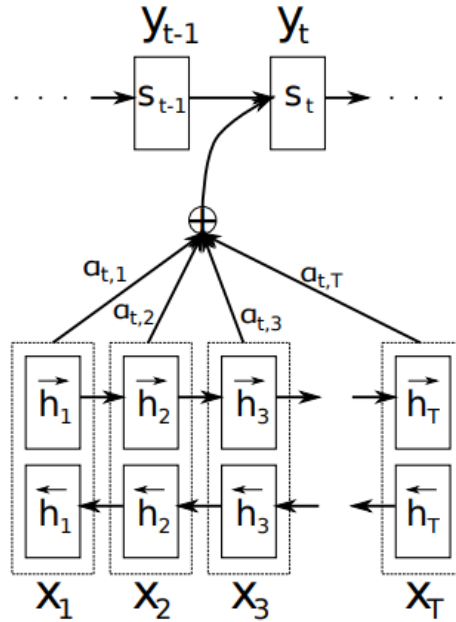


## অধ্যায় 10

# এটেনশন মেকানিজম

সিকুয়েন্স টু সিকুয়েন্স মডেলে একটি বটলনেক রয়েছে। রিকারেন্ট নেটের সর্বশেষ হিডেন স্টেট নিয়ে কেবল কাজ করতে পারবো আমরা। এর মানে আমাদের সকল ইনফর্মেশন একটা জায়গায় ফিক্সড। এই লাস্ট হিডেন স্টেটকে কন্টেক্সট ভেক্টর বলা হয়ে থাকে। ফিক্সড ভেক্টর ব্যবহার করলে নেটওয়ার্ক লং টার্ম ডিপেন্ডেন্সি নিয়ে কাজ করতে পারে না। সেই সমস্যা সমাধানে এটেনশন মেকানিজম ইউজ করা হয়। এমন যদি হয় যে ডিকোডার একটা ফিক্সড কন্টেক্সট না নিয়ে প্রতি টাইম স্টেপে এনকোডারের পুরো সিকুয়েন্স নিয়ে কাজ করতে পারবে? পুরো সিকুয়েন্সে যদি এনকোডার এক্সেস থাকে তাহলে এনকোডার যে স্টেটটি বেশি গুরুত্বপূর্ণ সেটিকে ব্যবহার করতে পারবে। এটেনশন ব্যবহার করলে এনকোডারের সবগুলি সিকুয়েন্সের উপর কাজ করা যায় এবং ইনপুটের সবচে উপযোগী অংশ মডেল ব্যবহার করতে পারে।

### 10.0.1 বাহদান এটেনশন



চিত্র 10.1: এটেনশন মেকানিজম

10.0.2 মাল্টি হেড এটেনশন

10.0.3 সেলফ এটেনশন

10.0.4 সেক২সেক এপ্লিকেশন

নিউরাল ইমেজ ক্যাপশনিং

10.0.5 ট্রান্সফর্মার

10.0.6 সেলফ এটেনশন ও কনভল্যুশনের মধ্যকার সম্পর্ক

## অধ্যায় 11

# রিকার্সিভ ও কনভল্যুশনাল নিউরাল নেটস ফর ল্যাঙ্গুয়েজ প্রসেসিং

### 11.0.1 রিকার্সিভ নিউরাল নেটস

### 11.0.2 কনভল্যুশন

কনভল্যুশন হলো এক ধরনের ফিচার একস্ট্রাকটর। কনভল্যুশন অপারেশনের মাধ্যমে আমরা ইনপুট টেক্সটের লোকাল ফিচারগুলি খুঁজে বের করতে পারি। কনভল্যুশন এক ধরনের গাণিতিক অপারেশন যার কাজ হল ইনপুট ম্যাট্রিক্সের উপর স্লাইডিং উইন্ডো ফাংশন এপ্লাই করা। স্লাইডিং উইন্ডো ফাংশনটির অনেক নাম, কার্নেল, ফিল্টার বা ফিচার এক্সট্রাক্টর। কনভল্যুশন বুঝতে গেলে আমাদের উইন্ডো সাইজের বিষয়টি খেয়াল করতে হবে। যেটা হচ্ছে আমরা একটা ফিল্টার নিছি (ধরা যাক  $3 \times 3$ , কার্নেল সাইজ = 3) এবং পুরো ছবির উপর স্লাইড করছি। একবার স্লাইড করলে আমরা ইনপুট ম্যাট্রিক্সের উপর একবার স্ট্যাম্প করবো। এখন আমাদের ইনপুটের একটা ব্লক আছে আর ফিল্টার আছে যা ইনপুটের ব্লকের উপর। এখানে আমরা এলিমেন্ট ওয়াইজ ম্যাট্রিক্স মাল্টিপ্লিকেশন করে তারপর যোগ করে নিবো।

### 11.0.3 কনভল্যুশন ফর সেনটেন্স মডেলিং

টেক্সটের জন্য সাধারণত 1D কনভল্যুশন ব্যবহার করা হয় [?]। ধরা যাক, আমাদের কাছে একটি বাক্য রয়েছে। আমরা উইন্ডো সাইজ নিলাম 2। 1D কার্নেলটি হবে  $(1 \times 3)!$  এখন আমরা এই দুই উইন্ডো সাইজ নিয়ে টেক্সটের উপর স্লাইড করতে পারি। প্রতিটা স্লাইডিং স্টেপে আমরা একটা নিউমারিক্যাল ভ্যালু পাবো। এভাবে পুরো বাক্য স্লাইড করা শেষ হলে আমরা একটা ফিচার পাবো যেটা কো-অকারেস ইনফর্মেশন সংগ্রহ করেছে। এখন যদি আমরা অনেকগুলি ফিচার ম্যাপ ব্যবহার করে তাহলে আমরা নানান রকম ফিচার এক্সট্রাক্ট করতে পারবো।

মজার বিষয়, আমরা চাইলে যেকোন উইন্ডো সাইজ ব্যবহার করতে পারি। সাধারণত উইন্ডো সাইজ হতে পারে 1,2,3। এখানে লক্ষ্য করলে বুঝা যাবে আমরা যখন স্লাইড করছি তখন কো-অকারেস কম্পিউটেশন হচ্ছে। মানে আমরা সেন্টার পয়েন্টের আশেপাশে কোন শব্দগুলি আছে সেগুলিকে ফিল্টার ম্যাপের আওতাও বিবেচনা করছি। ধরা যাক,  $3 \times 3$  ফিল্টার সাইজ, সেন্টার ওয়ার্ড হলো মাঝখানে শব্দটি আর ডান বাঁয়ে দুটি শব্দ আমাদের সেন্টার ওয়ার্ডের প্রতিবেশী শব্দ। আমরা যদি এক ধাপ স্লাইড করি তাহলে সেন্টার ওয়ার্ডটি তখন পরের ধাপে হয়ে যাবে প্রতিবেশী শব্দ। কনভল্যুশন কো-অকারেসের উপর নির্ভর করে আমাদেরকে ভালো স্প্যাশিয়াল ফিচার দিবে। মানে কোন শব্দ কার আশেপাশে বসতে পারে এই ইনফর্মেশন মডেল শিখবে ইনপুট থেকে। উদাহরণ বাক্য, আমার মন ভালো। মন শব্দের পর সাধারণত "ভালো", "খারাপ" এই শব্দগুলি নিয়মিত বসে। আমরা যদি অনেক ইনপুট নিয়ে কাজ করি মডেল এই কো-অকারেস বৈশিষ্ট্য শিখে নিতে পারবে। যদি এটা মডেল বুঝতে পারে তাহলে সে বাক্যের গঠন সম্পর্কে এক ধরনের লার্নিং পাচ্ছে যা ফিচার হিশেবে গুরুত্বপূর্ণ।

### n-গ্রাম ও 1D কনভল্যুশন

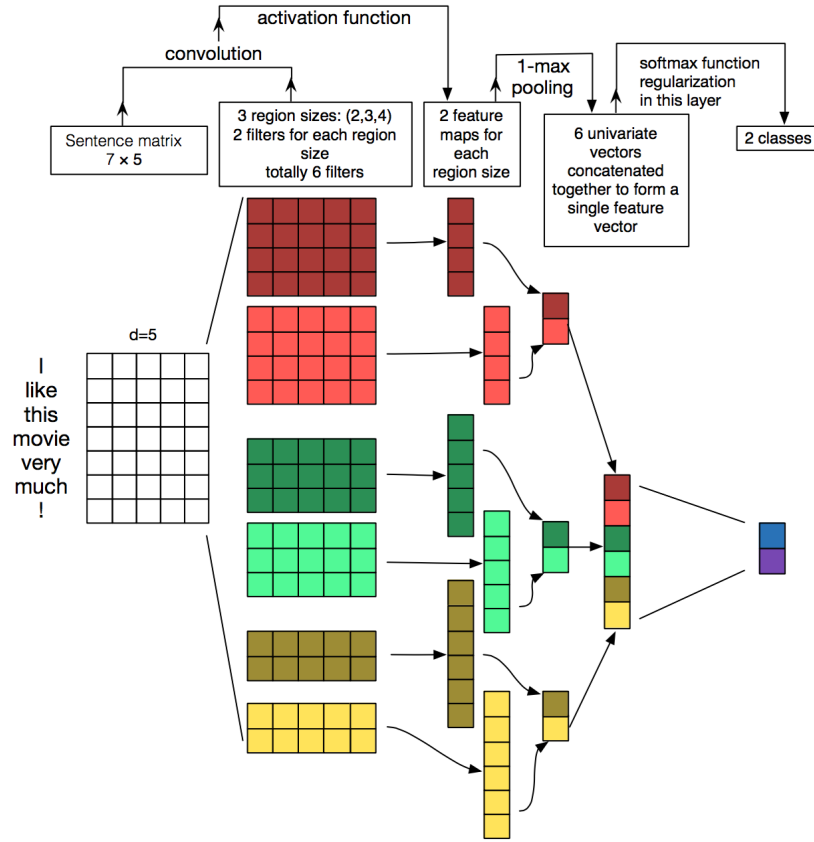
মন দিয়ে লক্ষ্য করলে বুঝা যাবে, কনভল্যুশন অপারেশনের সাথে n-gram মডেলের সাদৃশ্য আছে। লক্ষ্য করি, ফিল্টার সাইজ 2 হলে এটা একটা 2 গ্রাম মডেল কেননা মডেল ঠিক বাইগ্রামের মতন করে ফিচার তৈরি করবে। আমরা কনভল্যুশনকে n-গ্রাম ফিচার এক্সট্রাক্টর হিশেবে চিন্তা করতে পারি। এখন আমাদের পরের ধাপে কাজ হবে ফিচারগুলি জোড়া দিয়ে নেটওয়ার্ক ট্রেন করা।

কনভল্যুশনকে ন্যাচারাল ল্যাংগুয়েজ প্রোসেসিংয়ে phrase বেইজড মডেল হিশেবে ব্যবহার করা হয়। রিকারেন্ট নেটে সিকুয়েন্সিয়াল ডিপেন্ডেন্সি থাকে। রিকারেন্ট নেট prefix নিয়ে কাজ করে আর কনভল্যুশন নেটওয়ার্ক n-gram phrase নিয়ে কাজ করে। একটা বাক্য রিকারেন্ট নেটওয়ার্ক প্রসেস করবে সিকুয়েন্সিয়াল ডিপেন্ডেন্সি তৈরি করে আর কনভল্যুশন নেটওয়ার্ক অনেকগুলি phrase থেকে ফিচার নিবে। কনভল্যুশন নেটওয়ার্কের ইনপুট দিবে ওয়ার্ড ভেক্টরস। আমরা অনেকগুলি ফিল্টার ব্যবহার করতে পারি। সাধারণত ফিল্টার সাইজ 2,3,4,5 হয়ে থাকে। লক্ষ্য করি, ফিল্টার সাইজ 2 হলে এটা একটা 2 গ্রাম মডেল কেননা মডেল ঠিক বাইগ্রামের মতন করে ফিচার তৈরি করবে। আমরা কনভল্যুশনকে n-গ্রাম ফিচার এক্সট্রাক্টর হিশেবে চিন্তা করতে পারি।

## পুলিং ও ডিপ স্ট্যাকিং

এখন প্রশ্ন হচ্ছে কনভলুশন লেয়ারে অনেকগুলি ফিচার আমরা পাবো। সবগুলি ফিচার কি আমাদের কাজে লাগবে? উত্তর হচ্ছে, না! এইজন্য পরের লেয়ারে আমরা একটি পুলিং অপারেশন করবো। পুলিং অপারেশনের কাজ হলো সবচে ভালো ফিচার ভেক্টরগুলি খুঁজে বের করা। তারপর আমরা একটি সফটম্যাক্স লেয়ার ব্যবহার করে ক্লাসিফিকেশন কাজ করতে পারি। আমরা চাইলে কনভলুশন + পুলিং একটা পর একটা স্ট্যাক করতে পারি। ডিপ নিউরাল নেটওয়ার্ক স্ট্যাকিং ভালো ক্লাসিফিকেশন পারফরমেন্স দেয়।

## 11.0.4 কনভলুশন ফর সেন্টেন্স ক্লাসিফিকেশন



চিত্র 11.1: সেন্টেন্স ক্লাসিফিকেশনের জন্য কনভলুশন নিউরাল নেটওয়ার্ক আর্কিটেকচার ইলাস্ট্রেশন

## 11.0.5 কনভলুশন ড্রইনিংঃ কিছু প্র্যাক্টিক্যাল ট্রিকস

## 11.0.6 রিকারেন্ট ও কনভলুশনাল নিউরাল নেটওয়ার্কের তুলনামূলক আলোচনা

## অধ্যায় 12

# প্রিটেইন্ড ল্যাঙ্গুয়েজ মডেলস

### 12.0.1 কনটেক্সুলাইজড রিপ্রেজেন্টেশন্স

যদি আমরা ওয়ার্ড২ভেকে ওয়ার্ড অর্ডার রাখতে পারি তাহলে কি রিপ্রেজেন্টেশন আরো ভালো হবে না? আমরা অনেক ভালো সিম্যান্টিকস এবং সিনট্যাক্স ইনফরমেশন ম্যাশিনকে দিতে পারবো।

### 12.0.2 প্রি-টেইনিং

### 12.0.3 বার্ট (BERT)

### 12.0.4 জিপিটি ৩ (GPT 3)



খন্ড IV

পাট ৪





## অধ্যায় 13

# GLUE বৈশিষ্ট্যমূলক টাস্কস

গ্লু বৈশিষ্ট্যমূলক [?]

### 13.1 সেন্টেন্স ক্লাসিফিকেশন

### 13.2 ন্যাচারাল ল্যাঙ্গুয়েজ ইনফারেন্স

### 13.3 কোশেন আনসারিং

আজকে আমরা কোশেন আনসারিং নিয়ে আলাপ করবো। ইন্টারনেটে আমরা কিছু কোশেন করতে পারি যেমনঃ

- কখন প্রথম পিরামিড নির্মাণ করা হয়?
- সাকিব আল হাসান
- জয়া আহসানের বয়স কত?
- লন্ডন শহরের তাপমাত্রা কেমন?
- কেন আমরা ক্লান্ত লাগলে ঝিমাই?
- ক্রিস্টোফার নোলানের টপ ৫ টা মুভি

কোশেন অনেক রকমের হতে পারে এবং উত্তর বিভিন্ন ভাবে আসতে পারে। কোশেন আনসার ডোমেইন কিভাবে কাজ করে সেটা আমরা দেখবো।

প্রশ্ন	উত্তর
কখন প্রথম পিরামিড নির্মাণ করা হয়?	২৬৩০ BC
সাকিব আল হাসান	সাকিব আল হাসান একজন বাংলাদেশ ক্রিকেটার এবং অলরাউন্ডার। ২০০৬ সাল থেকে সাকিব বাংলাদেশ ক্রিকেট দলে খেলছেন।
জয়া আহসানের বয়স কত?	৪০ বছর
লন্ডন শহরের তাপমাত্রা কেমন?	৭ ডিগ্রী সেলসিয়াস। আকাশ পরিষ্কার, কিছুটা মেঘ দেখা যাচ্ছে।
কেন আমরা ক্লান্ত লাগলে ঝিমাই?	বিশ্রামের জন্য মন কাঁদে।

সারণী 13.1: প্রশ্ন উত্তর উদাহরণ।

কেন আমরা কোশেন আনসারিং করবো? অনলাইনে বিলিওনস ডকুমেন্ট আছে। এআই এর মৌলিক একটা গোল হলো সার্চিং। যেমন আপনি একটা ফ্লাইট বুক করতে চান। সেক্ষেত্রে আপনি চ্যাটবট ব্যবহার করে কোন হিউম্যান ইন্টারভেনশন ছাড়া ফ্লাইট বুক করতে পারছেন। যখন আমরা জিজ্ঞেস করছি জয়া আহসানের বয়স কত; তখন সিস্টেমকে বিলিয়নস উইকিপিডিয়া ডকুমেন্ট থেকে সার্চ করে উত্তর খুঁজে বের করতে হচ্ছে; যাকে বলার ইনফরমেশন একট্র্যাকশন। ক্রিস্টোফার নোলানের টপ ৫ টি মুভির নাম এবং মূল থিম। এটা হলো সামারাইজেশন টাস্ক। QA সিস্টেম প্রথমে ডকুমেন্টগুলি ওয়েব সার্চিং টুল দিয়ে র‍্যাংক করবে রিলেভেন্ট ডকুমেন্টগুলি। সেগুলি হতে পারে আইএমডিবি বা রটেন টমেটো কিছু ওয়েব পেইজ। তারপর সেগুলি থেকে ১০০০ হাজার শব্দকে সামারাইজ করে হয়ত ১০০ শব্দের

একটা সামারি তইরি করবে। লক্ষ্য করুন এখানে কোশ্চেন আনসারিং করতে ইনফরমেশন এক্সট্রাকশন এবং সামারাইজেশন দুইটা অতিরিক্ত কাজ করা লাগবে। কোশ্চেন আনসারিং সমস্যাকে বলা হয় এআই-কমপ্লিট। এর মানে হলো কোশ্চেন আনসারিং সমাধান করে ফেললে আমরা এআই এর অন্যান্য সমস্যাগুলিও সমাধান করে ফেলেছি। এছাড়া কোশ্চেন আনসারিং অনেকগুলি এপ্লিকেশন আছে। যেমন সার্চিং, ডায়ালগ, ইনফরমেশন একট্রাকশন, সামারাইজেশন। এই প্রত্যেকটা টুলই আমরা গুগোল সার্চ বা অন্যান্য ওয়েব সার্ভিস ব্যবহার করার সময় কাজে লাগাচ্ছি যার মূল ভিত্তি কোশ্চেন আনসারিং। প্রথম সফল কোশ্চেন আনসারিং সিস্টেম ধরা হয় আইবিএম জিওপার্ডিকে। যদিও পরে যতোটা ঢাকঢোল পিটানো হয়েছিল আইবিএম সেসব সত্যি কাজ করে নি। আমার দৈনন্দিন জীবনে সার্চিং, ফ্লাইট রিজার্ভেশন, পণ্য অর্ডার করা, সিরি এগুলি নিয়মিত সফল QA সিস্টেম। এছাড়াও আরো বিভিন্ন রকমের সমস্যা আছে। যেমন একটা প্রশ্ন হতে পারেঃ সাকিব আল হাসান কতগুলি টি টুয়েন্টি ম্যাচে ৫০ করেছেন। এই কাজের জন্য ম্যাশিনকে একটা টেবুলার ডাটা থেকে কাউন্টিং করে উত্তর খুঁজে বের করতে। তাহলে আমরা দেখতে পাচ্ছি QA সিস্টেম গাণিতিক সমস্যা সমাধান করার প্রয়োজন পড়ে। আরেকটি কঠিন সমস্যা হলো রিজনিং। যেসব প্রশ্নের উত্তর ডিরেক্টলি দেয়া সম্ভব না। যেগুলি উত্তর করতে গেলে রিজনিং করে উত্তর দিতে হয়। যেমনঃ যেকোন ধরনের মোরাল এথিক্যাল কুশ্চেন। কেউ যদি সিরিকে প্রশ্নকে আমার মা আমাকে বকা দিয়েছে, আমি কি নিজেকে কষ্ট দিব ওর উপর রাগ করে? এই ধরনের প্রশ্নের উত্তর দিতে গেলে এআই সিস্টেমকে হিউম্যান ইন্টেলিজেন্সের আন্স্পেক্টগুলি যেমন মোরালিটি, এথিকস, রিজনিং, মিনিং এগুলি বুঝতে হবে। QA আমরা নিত্যকার জীবনে রোজ ব্যবহার করছি। সিরি প্রশ্ন উত্তর, গুগোল সার্চের একটা বড় অংশ QA সিস্টেম।

প্রশ্ন	উত্তরের উৎস
কখন প্রথম পিরামিড নির্মাণ করা হয়?	এনসাইক্লোপিডিয়া
সাকিব আল হাসান	সাম্প্রতিক এনসাইক্লোপিডিয়া / উইকিপিডিয়া
জয়া আহসানের বয়স কত?	পত্রিকা / উইকিপিডিয়া
লন্ডন শহরের তাপমাত্রা কেমন?	আবহাওয়া দফতর
কেন আমি পরীক্ষায় ফেল করেছি	ব্যক্তিগত অবজার্ভেশন
কেন আমরা ক্লান্ত লাগলে কিমাই?	বিভিন্ন গবেষণাপত্র

সারণী 13.2: প্রশ্ন এবং উত্তরের বিভিন্ন উৎস।

প্রথম দুটি উত্তর পাবো উইকিপিডিয়া থেকে। স্টক মার্কেটের তথ্য স্টক একচেঞ্জ ড্যাটাবেজ থেকে। আবহাওয়ার খবর আবহাওয়া দফতর থেকে। নিজের ব্যক্তিগত অবজার্ভেশন থেকে উত্তর পাচ্ছি। কিছু ক্ষেত্রে বিভিন্ন গবেষণা পত্র পড়ে উত্তর খুঁজে নিতে পারি। কোশ্চেন আনসারিং সিস্টেম ৩ রকমের ড্যাটার উপর নির্ভর করে।

প্রশ্ন	কন্টেন্ট / উৎস	উত্তর
ফ্যাকচুয়াল প্রশ্ন	একগুচ্ছ ডকুমেন্ট (করপাস)	একটাই ফ্যাক্ট
জটিল / বর্ণনামূলক প্রশ্ন		একটা এক্সপ্ল্যানেশন
ইনফরমেশন রিট্রাইভাল	একটি ডকুমেন্ট	নলেজ বেইজ
		বাক্য কিংবা প্যারাগ্রাফ কোথাও থেকে এক্সট্রাক্ট করা
	নন টেক্সচুয়াল ডাটা (ছবি, সেন্সর, জিপি-এস)	একটি ছবি বা অন্যান্য অবজেক্ট
		আরেকটি প্রশ্ন

সারণী 13.3: প্রশ্ন এবং উত্তরের বিভিন্ন উৎস।

এই রকমফেরের উপর নির্ভর করে আমরা বিভিন্ন রকম QA সিস্টেম ডিজাইন করার কথা ভাবতে পারি। আমরা দেখতে পাচ্ছি প্রশ্ন ফ্যাকচুয়াল (অনেকগুলি ডকুমেন্ট খুঁজে একটাই উত্তর; হ্যাঁ/ না, বহর), বর্ণনামূলক কুশ্চেন (একটা করপাস ঘেটে একটা ব্যাখ্যা তইরি) এবং ইনফরমেশন রিট্রাইভাল (একটা ডকুমেন্ট, কিংবা নলেজ বেইজ থেকে বা ছবির ড্যাটাবেজ একটা ছবি বের করা)। কমপ্লেক্স বা ন্যারেটিভ প্রশ্নের উদাহরণ হতে পারে আমার কম্পিউটার ব্লু স্ক্রিন কেন হয়ে যায় মাঝে মাঝে এবং এটা কিভাবে ফিক্স করতে পারি? আরেকটি জনপ্রিয় উৎস হচ্ছে নলেজ বেইজ যা অনেক পুরান এনএলপি রিসার্চ টেকনিক। এটা এখনো অনেক একটিভ এবং কঠিন রিসার্চ এরিয়া। অনেক সময় প্রশ্নের উত্তরে পাল্টা প্রশ্ন করা হয় নির্দিষ্ট প্রশ্ন বা নিশ্চিত হবার জন্য। সাধারণত ডায়ালগ সিস্টেমে সেই ধরনের পাল্টা প্রশ্ন উত্তর ইন্টারফেস থাকে। যেমন আমি যদি হোটেল বুকিং সিস্টেমে প্রশ্ন করি আগামী ৭ দিনে মন্ট্রিয়ল শহরে কোন হোটেল পাওয়া যাবে কিনা? তখন সিস্টেম প্রশ্ন করতে পারে আমার কোন পছন্দের এলাকা আছে কিনা? আমার উত্তরের সাপেক্ষে সার্চ আরো ন্যারো ডাউন করার জন্য সিস্টেম আমাকে জিজ্ঞেস করতে পারে আমার কয়টা রুম লাগবে বা কজন থাকবো? এটা হলো প্রশ্নের উত্তর পাল্টা প্রশ্ন যেগুলি নিয়মিত আমরা নিত্যদিন ব্যবহার করছি।

প্রশ্নের ট্যাক্সোনোমি

অনেক রকম ট্যাক্সোনোমি হতে পারে প্রশ্নের জন্যঃ

- ☐ কি কেন কখন শব্দ দিয়ে প্রশ্ন (Wh- কুশ্চেন)
- ☐ প্রশ্নের বিষয়
- ☐ উত্তরের আকার (form)
- ☐ উৎসের টাইপ যেখান থেকে উত্তর পাওয়া যেতে পারে

এই ট্যাক্সোনোমিগুলি মাথায় রেখে আমরা প্রশ্ন উত্তর সিস্টেম বানাতে পারি। যেমন আমাদের স্কুলে যে রিডিং কম্প্রিহেনশন থাকতো সেখানে উত্তর অই নির্দিষ্ট প্যাসেজ থেকেই খুঁজে বের করতো হতো? তাহলে আমরা যদি এমন একটা প্রশ্ন উত্তর সিস্টেম বানাতে চাই যেটা রিডিং কম্প্রিহেনশন করতে পারবে আমাদের ম্যাশিন লার্নিং টাস্কটা ওয়েল ডিফাইনড হবে। তাহলে কি ধরনের উৎস থেকে উত্তর পাচ্ছি সেটার উপর নির্ভর করে QA টাস্ক তৈরি করা এবং ড্যাটাসেট এভালুয়েশন এগুলি সহজ হয়। কাজের সুবিধার্থে সাধারণত প্রশ্ন উত্তর সিস্টেম বানানোর জন্য উত্তরের উৎস চিন্তা করে আমরা কাজ শুরু করে দিতে পারি। কারণ ড্যাটা সোর্স ম্যাশিন লার্নিং ডিজাইনের জন্য সবচেয়ে ক্রিটিক্যাল উপাদান। বলা যেতে পারে, আমরা প্রশ্নের চেয়ে উত্তরের উপর বেশি ফোকাস করছি।

এখন আমাদের কাছে প্রশ্ন উত্তর সিস্টেম পরের ধাপে ৩ টি প্রশ্ন মাথায় রাখতে হবে:

- ☐ উত্তর দেখতে কেমন হবে। যেমনঃ হ্যাঁ/না, ড্যাটাবেইজ লুক আপ, প্যাসেজ থেকে একটা বাক্যাংশ ইত্যাদি।
- ☐ আমরা উত্তর কোথায় পাবো (উত্তরের উৎস)? যেমনঃ উইকিপিডিয়া, রিডিং প্যাসেজ, ড্যাটাবেইজ টেবিল ইত্যাদি।
- ☐ আমার ট্রেনিং ড্যাটা দেখতে কেমন হবে (ম্যাশিন লার্নিং টাস্ক)? একটা প্যাসেজ থাকবে এবং প্যাসেজ থেকে কিছু রিলেভেন্ট প্রশ্ন। উত্তর হবে প্যাসেজের বাক্যাংশ।

আমরা এখন দেখবো লিটারেচারে কিভাবে কোশ্চেন আনসারিং টাস্ক কিভাবে ডিফাইন করা হয়েছে। নিচে কয়েকটি কোশ্চেন আনসারিং এরিয়া লিস্ট ডাউন করা হলো:

রিডিং কম্প্রিহেনশন	<input type="checkbox"/> উত্তর একটা ডকুমেন্ট থেকে পাবো <input type="checkbox"/> কন্টেন্ট হলো ডকুমেন্টটি
সিমান্টিক পার্সিং	<input type="checkbox"/> উত্তর একটি লজিক্যাল ফর্ম, একটা নজেল বেইজে এক্সিকিউট করা যায় <input type="checkbox"/> কন্টেন্ট হলো নজেল বেইজ
ভিজুয়াল QA	<input type="checkbox"/> উত্তর সহজ এবং ফ্যাকচুয়াল <input type="checkbox"/> উত্তর এক বা একের অধিক ছবি
ইনফরমেশন রিট্রাইভাল	<input type="checkbox"/> উত্তর একটি ডকুমেন্ট / প্যারাগ্রাফ / বাক্য <input type="checkbox"/> কন্টেন্ট একটি ডকুমেন্টের করপাস

রিডিং কম্প্রিহেনশন আমরা উচ্চ মাধ্যমিকে পড়েছি। একজ্যাক্টলি একই কাজ আমরা এখন ম্যাশিনকে দিয়ে করাবো। এবার ধরা যাক আমাদের ড্যাটাবেইজে একটা টেবিল আছে যেখানে প্রতি সারিতে বাংলাদেশ খেলোয়াড়রা কোন বছরে কতোগুলি ৫০ করেছেন তার তথ্য আছে। এটা একটা স্ট্রাকচারড ড্যাটা সোর্স। এখন আমরা আগে যে প্রশ্নটি করলাম সাকিব কতোগুলি ৫০ করেছেন? এই প্রশ্নটি একটা ন্যাচারাল কোশ্চেন। আমরা এটা ডিরেক্টলি ড্যাটাবেইজ কুয়েরি করতে পারবো না। এটা উত্তর পেতে আমরা প্রথমে ন্যাচারাল ল্যাঙ্গুয়েজটাকে SQL এ রূপান্তর করবো এবং তারপর ড্যাটাবেইজ কুয়েরি করতে পারবো। এই টাস্কটিকে ম্যাশিন লার্নিংয়ে বলা হয় সিমান্টিক পার্সিং। এটা এন্টারপ্রাইজ সল্যুশন বিশেষত ব্যাংকিং এ প্রচুর ব্যবহৃত হয়। ন্যাচারাল ল্যাঙ্গুয়েজ থেকে যেকোন ড্যাটাবেইজ কুয়েরি রূপান্তর করাই সিমান্টিক পার্সিং। ভিজুয়াল কুশ্চেন আনসারিংয়ে আমাদের একটা ভিশন সিস্টেম থাকবে এবং একটা ল্যাংগুয়েজ সিস্টেম থাকে। ভিশন সিস্টেম ইমেজ প্রসেস করে এবং ন্যাচারাল ল্যাংগুয়েজ সিস্টেম ইমেইজের মধ্যে যে তথ্য বা অবজেক্টগুলি সেগুলি নিয়ে প্রশ্ন করলে; সেটা উত্তর হিসেবে দেয়। যেমন নিচের ছবিতে একটা প্রশ্ন করতে পারি আমরা কয়টা হলুদ রঙের কলা আছে? অন্যান্য QA টাস্কের তুলনায় ভিজুয়াল QA তুলনামূলক বেশি কঠিন কেননা একই সাথে ভিশন এবং ল্যাংগুয়েজ দুটি সমস্যা নিয়ে কাজ করতে হয়। ইনফরমেশন রিট্রাইভাল হলো ক্লাসিক ড্যাটা মাইনিং সমস্যা। আমাদের নেইম এন্টিটি একটা ইনফরমেশন রিট্রাইভাল টাস্ক। ওয়েব ডকুমেন্ট সার্চিং আরেকটি উদাহরণ।

### 13.3.1 রিডিং কম্প্রিহেনশন

### 13.3.2 ওপেন ডোমেইন কুশ্চেন আনসারিং



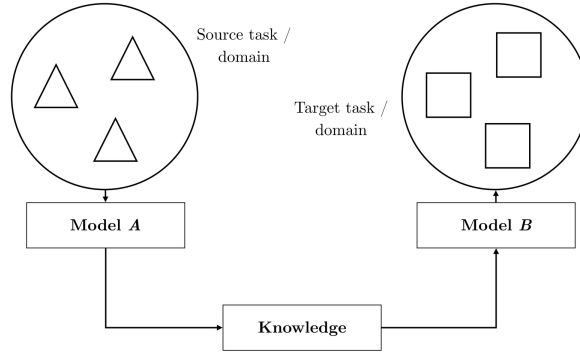
## অধ্যায় 14

# ট্রান্সফার লার্নিং ফর ল্যাঙ্গুয়েজ প্রসেসিং

এনএলপি তে কিছু রিয়েল লাইফ চ্যালেঞ্জ রয়েছে। সাধারণ ডিপ লার্নিং সিস্টেম ট্রেন করলে অনেক ডাটা লাগে। ধরা যাক, কুশ্চেন আনসারিং সিস্টেম বা ম্যাশিন ট্রান্সলেশন। আমাদের কয়েক হাজার ট্রেনিং স্যাম্পল লাগবে একটা ভালো সিস্টেম ট্রেন করতে। অনেক ডিপ লার্নিং সমস্যা সমাধানে যথেষ্ট ডাটা পাওয়া যায় না। ধরা যাক, ৫০০০০ এর বেশি ট্রেনিং স্যাম্পল লাগছে ট্রেনিং করানোর জন্য। আরেকটা বিষয় হচ্ছে সোর্স এবং টার্গেট ডিস্ট্রিবিউশন সমান হতে হবে। এই সমস্যার সহজ সমাধান হচ্ছে টার্গেট ডাটা নিয়ে আরেকটি মডেল ট্রেন করানো। এবং টার্গেট ডোমেইনে ল্যাবেলড ডাটা পাওয়া মুশকিল। আমরা দেখা গেলো একটা ডাটাসেট আমরা নর্থ আমেরিকান কনটেন্টে বানালাম। সেটা এশিয়ান এই সমস্যাগুলি সমাধানে ট্রান্সফার লার্নিং ব্যবহার করা হয়। কিন্তু অন্য কোন সমস্যা সমাধান করতে গিয়ে আমরা যা শিখেছি সেটা আমরা কাজে লাগাতে পারি। এটাকে বলা হয় ট্রান্সফার লার্নিং। শব্দ দুটি দেখেই বুঝা যাচ্ছে লার্নিং কে আরেকটা কাজে ট্রান্সফার করা হবে।

প্রিট্রেন্ড ল্যাঙ্গুয়েজ মডেলের জন্য ট্রান্সফার লার্নিং খুবই নিয়মিত এখন। ট্রান্সফার লার্নিং অনেকগুলি টপিক আছে। আমাদের কিছু এজাম্পশন দিয়ে কাজ আগাতে হবে। প্রথমঃ আমার টার্গেটে ল্যাবেলড ডাটা আছে (ইনডাক্টিভ ট্রান্সফার লার্নিং), ল্যাবেলড ডাটা কেবল সোর্সেই আছে (ট্রান্সডাক্টিভ ট্রান্সফার লার্নিং)। আরেকটা হচ্ছে সোর্স এবং টার্গেট ল্যাবেলড ডাটা নেই (আনসুপারভাইজড ট্রান্সফার লার্নিং)। ডোমেইন এডাপ্টেশন হল ট্রান্সডাক্টিভ ট্রান্সফার লার্নিং এর একটি উদাহরণ। ডোমেইন এডাপ্টেশনের একটা উদাহরণ হতে পারে আপনার NER টাস্ক এর জন্য ইংরেজি ট্রেনিং ডাটা আছে কিন্তু বাংলায়। দুটি আলাদা ভাষা ডোমেইন আলাদা করেছে। এখানে টাস্ক সেইম। আমাদের কাজ হচ্ছে ইংরেজি ডোমেইন থেকে বাংলায় নলেজ ট্রান্সফার করা। কোভারিয়েট শিফট হলো অপটিমাল প্রেডিক্টর ফিক্সড, কিন্তু ফিচার স্পেস ডিফারেন্ট যেখানে সোর্স আর টার্গেট টাস্ক একই। পার্টস অব স্পিচ ট্যাগিং এবং নেইম এন্টিটি দুটি টাস্ক অনেকখানি সিমিলার। এনইআর ক্ষেত্রে লো লেভেল সিনট্যাক্টিক ইনফরমেশন দরকার হয়। পার্টস অব স্পিচ ট্যাগিংয়েও সিনট্যাক্স ফিচার গুরুত্ব বহন করে। ব্যাপারটা এমন যে আপনি যদি POS ট্যাগ জানেন কোন একটা বাক্যের তাহলে সেগুলি ব্যবহার করে NER করতে প্রচুর হিউরেস্টিকস পাওয়া যায়। POS ট্যাগের তথ্য NER টাস্কটি সহজ করে দেয়। আমাদের এই দুটি টাস্কের জন্য ল্যাবেলড ডাটা আছে। আমরা যদি এই দুটি টাস্ক একসাথে ট্রেন করাই এবং প্যারামিটার শেয়ার করে দিই তাহলে POS যা শিখছে এবং NER যা শিখছে দুটি টাস্ক একটা আরেকটাকে শেয়ার করতে পারে। এটাকে বলা হয় ইন্ডাক্টিভ ট্রান্সফার লার্নিং। ইন্ডাক্টিভ লার্নিং মানে হলো ইমপ্লিচিট লার্নিং। মানে আপনি যখন একের অধিক টাস্ক একসাথে শিখছেন তখন একে অপরকে অটোমেটিক্যালি সাহায্য করবে, কোন এক্সপ্লিচিট অবজেক্টিভ ছাড়াই। এটা লার্নিং প্রসিডিউর নিজে নিজেই এই প্রোপারটি এমার্জ করে। ইনডাক্টিভ ট্রান্সফার লার্নিংয়ের চমৎকার উদাহরণ হলো মাল্টি টাস্ক লার্নিং। কিন্তু ট্রান্সডাক্টিভ ট্রান্সফার লার্নিংয়ে আমরা টার্গেট ডোমেইনে এক্সপ্লিচিটলি নলেজ ট্রান্সফার নিয়ে অবজেক্টিভ যেমন লস ফাংশন, আর্কিটেকচার ডিজাইন করি। আনসুপারভাইজড লার্নিং ক্ষেত্রে আমরা লেবেলড ডাটা থেকে শেখার সুযোগ নাই। আমরা বলেছি আগে প্রবাবিলিস্টিক ম্যাশিন লার্নিংয়ের উদ্দেশ্য হলো ডাটা জেনারেটর প্রসেস শেখা বা এপ্রোক্সিমেট করা। ধরা যাক, আপনার কাছে d-ডাইমেনশনাল ডাটা আছে। এবং এটা প্লট করলে আপনি একটা হিস্টোগ্রাম পাবেন। ডেনসিটি এস্টিমেশনের কাজ হলো এই হিস্টোগ্রাম প্লটটা এপ্রোক্সিমেট করা। কারণ মূলত আপনি ডাটা জেনারেটিং ডিস্ট্রিবিউশন জানেন না। আপনার কাছে আছে কিছু স্যাম্পল ডাটা সেগুলি থেকে আপনি স্ট্রাকচারটা এস্টিমেট করে নিচ্ছেন ম্যাশিন লার্নিং মডেল দিয়ে। রিয়েল ওয়ার্ল্ড ডাটা ডিস্ট্রিবিউশন সাধারণত হাইডাইমেনশনাল এবং কমপ্লেক্স। এমন কমপ্লেক্স ডিস্ট্রিবিউশন জানা যায় না অধিকাংশ ক্ষেত্রে। আমরা জানি কয়েন ফ্লিপিং হলো বানুলি ডিস্ট্রিবিউশন। ডিস্ট্রিবিউশন জানা থাকলে ওখান থেকে আমরা ডাটা স্যাম্পল করতে পারবো। বানুলি  $\gamma$  ভালু জানলে, আমরা কয়েন ফ্লিপ হেড টেইল জেনারেট করতে পারবো। আমাদের ম্যাশিন লার্নিং টাস্ক হলো এই ডিস্ট্রিবিউশন এপ্রোক্সিমেট করা। যখন আমরা ডিস্ট্রিবিউশন জানবো, তখন আমরা একইসাথে ডাটা জেনারেটিং প্রসেস ইমিটেট করতে পারবো। আমরা যে দুটি ট্রান্সফার লার্নিং সমস্যা নিয়মিত ম্যাশিন লার্নিং কম্যুনিটিতে পাই একটি হলো ডোমেইন এডাপ্টেশন আরেকটি মাল্টিটাস্ক লার্নিং।

ট্রান্সডাক্টিভ ট্রান্সফারের কিছু বৈশিষ্ট্যঃ i) টার্গেটে ডোমেইনে কোন ল্যাবেলড ডাটা নেই ii) এনএলপি ট্রান্সফার লার্নিং রিসার্চ মোস্টলি এই টপিকে হয় এবং iii) উদাহরণ ডোমেইন এডাপ্টেশন। ইনডাক্টিভ ট্রান্সফার লার্নিংয়ের বৈশিষ্ট্যঃ i) টার্গেটে ল্যাবেলড ডাটা আছে ii) লক্ষ্য হলো অন্য টাস্কের সাহায্য নিয়ে টার্গেট টাস্কের পারফরমেন্স বাড়ানো। iii) একসাথে বিভিন্ন টাস্ক শিখানো (মাল্টিটাস্ক লার্নিং), প্রি-ট্রেনিং (ওয়ার্ড এম্বেডিংস)। যেমন সিম্যান্টিক সিমিলারিটি শেখা ওয়ার্ড এম্বেডিংসের অবজেক্টিভ ফাংশনে নেই। লার্নিং থেকে এই প্রোপারটি এমার্জ করে। আরেকটা উদাহরণ হতে পারে BERT ট্রেনিং। সিনট্যাক্টিক ফিচার শেখা প্রিট্রেনিংয়ের অবজেক্টিভ না। প্রিট্রেনিং অবজেক্টিভ হলো মাস্কড ল্যাংগুয়েজ মডেলিং। আমরা যখন BERT ট্রেন করছি তখন ব্যাগ অব ওয়ার্ডস, সিনট্যাক্স এগুলি শেখাই না তবু প্রথম দিককার লেয়ারগুলি এগুলি শিখে নেয়। এটাই হলো ইন্ডাক্টিভ ট্রান্সফার। এগুলি সে শিখতে পারে ল্যাংগুয়েজ মডেলিং শিখতে গিয়ে।



চিত্র 14.1: ট্রান্সফার লার্নিং

### 14.0.1 সিকুয়েন্সিয়াল ট্রান্সফার

#### প্রি-ট্রেনিং

ওয়ার্ড এম্বেডিংস প্রি ট্রেন্ড ওয়ার্ড এম্বেডিংস অধিকাংশ ডিপ লার্নিং মডেলের প্রথম লেয়ারে থাকে। প্রি ট্রেন্ড এম্বেডিং কিছু সমস্যা আছে। এটা শ্যালা এপ্রোচ। ওয়ার্ড এম্বেডিংস কনটেক্সট সেনসিটিভ না। এবং এগুলি সেস ধরতে পারে না। সাধারণত প্রথম লেয়ার প্রিট্রেন্ড করা থাকে, বাকি লেয়ারগুলি শুরু থেকে ট্রেন করতে হয়। এনএলপি প্রিট্রেনিং যে নতুন প্যারাডাইম শিফট সেটা এসছে ইমেজনেট প্রিট্রেনিং কম্পিউটার ভিশন থেকে। যেমন VggNet, ResNet। আমাদের কাছে যদি একটা ইমেজ থাকে তাহলে আমরা সেটা VGGnet এ ইনপুট হিসেবে দিলে সে একটা আউটপুট ফিচার দিবে। সেই ফিচার দিবে আমরা সহজেই ক্লাসিফাইয়ার ট্রেন করাতে পারবো। ভিশনে এখন প্রিট্রেন্ড মডেল ইমেজনেট একটা স্ট্যান্ডার্ড। প্রথম লেয়ারে এজ, তারপর টেকচার, প্যাটার্ন, তারপর পার্টস এবং শেষ লেয়ারে অবজেক্ট থাকে। এখন আমরা যেই টাস্কটি সলভ করতে চাই হয়ত সেটাই অবজেক্ট রিকগনিশন নয়। কিন্তু টেকচার সেই টাস্কের জন্য গুরুত্বপূর্ণ। সেক্ষেত্রে আমরা শুরু থেকে টেকচার লেয়ার কে প্রিট্রেন্ড লেয়ার হিসেবে নিতে পারি এবং পরে কিছু লেয়ার জোড়া দিয়ে একটা নতুন মডেল বানাতে পারবো। আধুনিক ম্যাশিন লার্নিংয়ে প্রিট্রেনিং ট্রান্সফার লার্নিংয়ের একটা key কনসেপ্ট হিসেবে কাজ করে। এখন প্রশ্ন হচ্ছে ল্যাঙ্গুয়েজ প্রসেসিংয়ে ইমেজনেট টাস্কের মতন টাস্ক কি হতে পারে? ইমেজনেট থেকে আমরা কিছু আইডিয়া নিতে পারি। এনএলপি ইমেজনেট ডাটাসেটটি যথেষ্ট বড় হতে হবে যেমন মিলিয়নস অব ট্রেনিং স্যাম্পল। এটা সাধারণ প্রব্লেম স্পেসকে রিপ্রেজেন্ট করতে হবে। এই ধরনের কিছু টাস্ক যদি চিন্তা করি তাহলে পাবোঃ i), রিডিং কম্প্রিহেনশন (SQuAD ডাটাসেট, 100k Q-A জোড়া) ii) ন্যাচারাল ল্যাংগুয়েজ ইনফারেন্স (SNLI করপাস, 570k বাক্য জোড়া) iii) ম্যাশিন ট্রান্সলেশন (WMT 2014, 40M ফ্রেঞ্চ - ইংরেজি বাক্য জোড়া) iv) ল্যাংগুয়েজ মডেলিং (অফুরন্ত ডাটা, এখন বেঞ্চমার্ক ডাটাসেট সাইজঃ 1B শব্দ)। কনসেপ্টটি হচ্ছে আপনার কাছে যদি বড় কোর্পাস আনসারিং ডাটাসেট থেকে ট্রেন্ড করা মডেল থাকে তাহলে সেই মডেলকে আপনি অন্য টাস্ক সলভ করার জন্য ব্যবহার করতে পারেন। কারণ ডাটাসেট সাইজ অনেক বড় এবং ট্রেনিং থেকে যে রিপ্রেজেন্টেশন লেয়ার পাওয়া যায় সেগুলি অন্য টাস্ককে সাহায্য করে। উপরের প্রথম ৩ টি টাস্ক ল্যাবেলড ডাটা দরকার পড়ে এবং ল্যাবেলড ডাটা সবসময় সীমিত। শেষ টাস্কে কোন ল্যাবেলড ডাটা লাগে না, যেমন 1B শব্দ আমরা ব্যবহার করতে পারি তাই এই ধরনের লার্নিং সিস্টেম ভালো প্রিট্রেনিং ফলাফল দেয়।

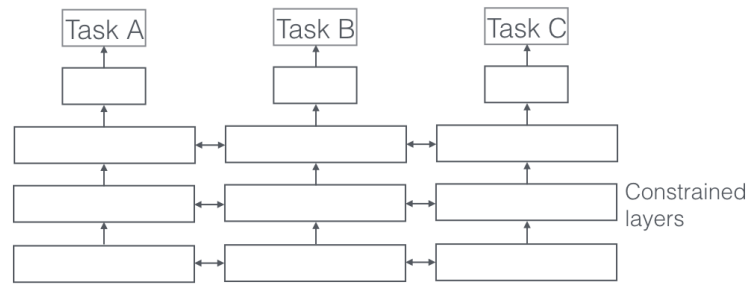
ল্যাংগুয়েজ মডেলিং ভাষার বেশ কয়েকটি ক্যাপচার আকর্ষণ করে। যেমন লংটার্ম ডিপেন্ডেন্সি, হায়ারার্কিক্যাল সম্পর্ক, সেন্টিমেন্ট ইত্যাদি। যেমন আমরা যে অবজেক্টিভ ফাংশন ব্যবহার করে তার মধ্যে আগের কন্টেক্সট এর উপর নির্ভর করে পরের বাক্য প্রেডিক্ট করতে তাই। এটা লং টার্ম ডিপেন্ডেন্সি ছাড়া সম্ভব নয়। সেন্টিমেন্ট আমরা ওয়ার্ড এম্বেডিংস দেখতে পাই খুব সহজেই। কাছাকাছি অর্থ বহন করে এমন শব্দ একই ক্লাস্টারে থাকে। ট্রেনিং ডাটা তো আনলিমিটেড থাকছেই।

#### ULMFiT

এই সেকশনে আমরা সিকুয়েন্সিয়াল ট্রান্সফার লার্নিং নিয়ে একটা ম্যানুয়াল আলোচনা করতে চাই। পেপারটির শিরোনাম হল Universal Language Model Fine-Tuning (ULMFiT)। আমাদের যে এপ্রোচ সেটি ইন্ডাক্টিভ ট্রান্সফার লার্নিং।

### 14.0.2 মাল্টি-টাস্ক ট্রান্সফার লার্নিং

### 14.0.3 লাইট ওয়েট ফাইন-টিউনিং



চিত্র 14.2: মাল্টি টাস্ক লার্নিং





## অধ্যায় 15

# এথিকস ও ন্যাচারাল ল্যান্ডস্কেপ প্রসেসিং

15.0.1 জেন্ডার বায়াস ও ওয়ার্ড এসেম্ভিং

15.0.2 ডিবায়াসিং

15.0.3 স্টকাস্টিক প্যারটস



## অধ্যায় 16

# ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং রিসার্চ

- 16.0.1 থিম ১ঃ ল্যাঙ্গুয়েজ মডেলিং
- 16.0.2 থিম ২ঃ কমনসেন্স রিজনিং
- 16.0.3 থিম ২ঃ মডেল এনালাইসিস ও এক্সপ্লোরেশন
- 16.0.4 থিম ৩ঃ মাল্টিলিঙ্গুয়াল ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং
- 16.0.5 থিম ৪ঃ কন্টিনিউয়াল লার্নিং