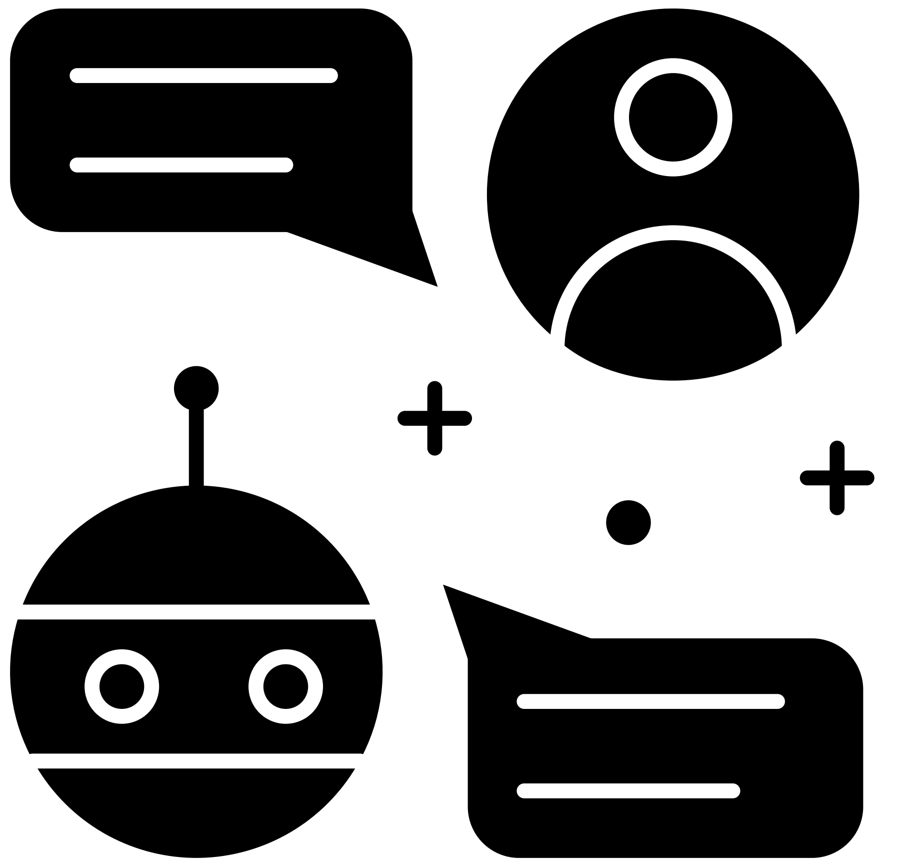
# Course: Build a Small Language Model from scratch

Module 1: Foundations and Setup



Instructor: Nusrat Jahan Lia

## Learning Objectives

By the end of this section, you will:

- Understand what language models are and how they work
- Differentiate between small and large language models
- Know the specific goals of our Bangla SLM project

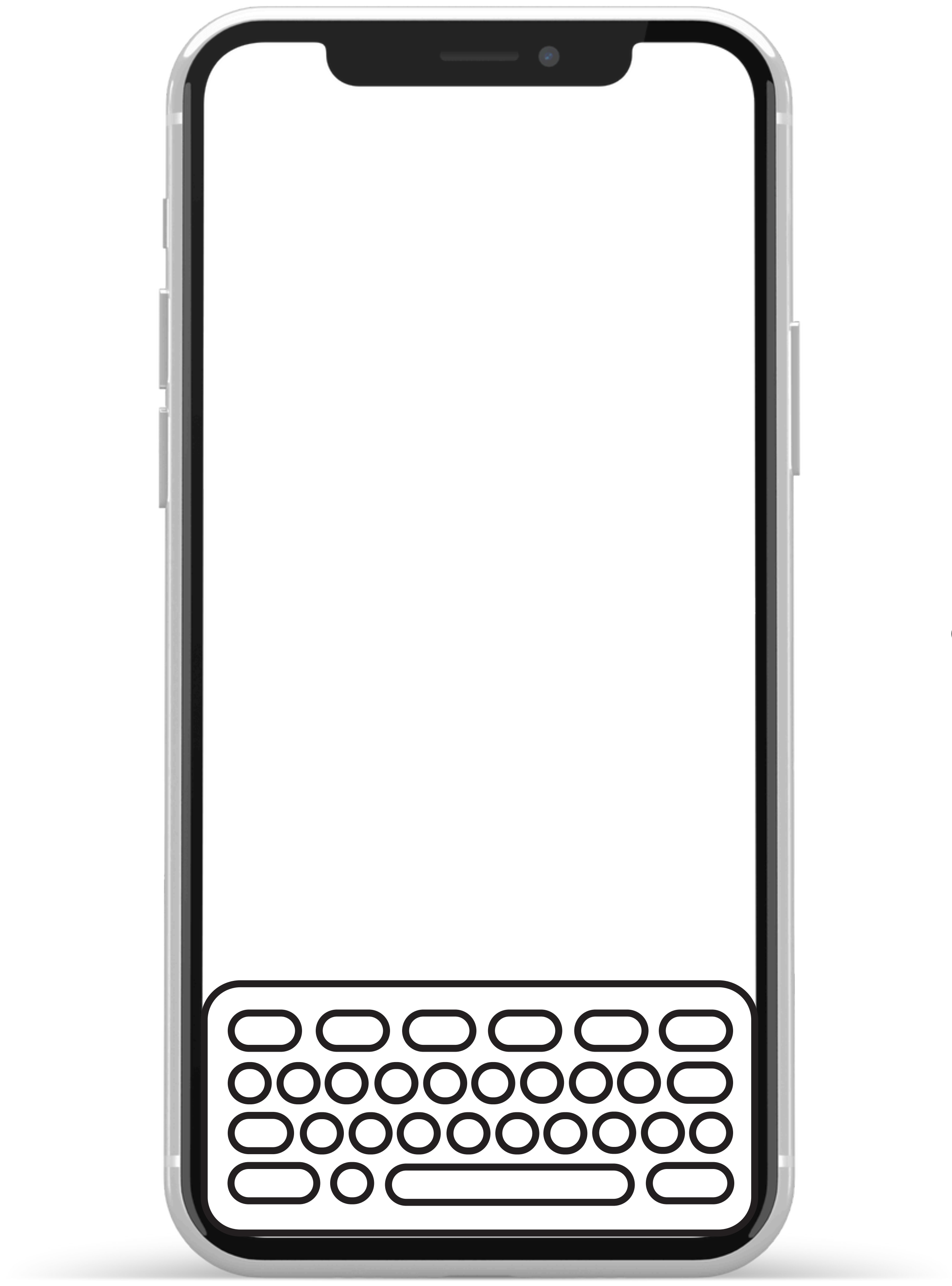
## 1.1 Introduction to Language Models

### What is a Language Model?

A language model is a computer program that learns to predict the next word in a sentence based on the words that came before it.

Think of it like an extremely sophisticated autocomplete system.

## Real-World Examples



When you type "Good morning, how are..." your phone suggests "you" as the next word

## Real-World Examples

- **Gmail Smart Compose:** Suggests completions for your emails
- **Google Search:** Autocompletes your search queries
- **ChatGPT:** Generates entire conversations by predicting one word at a time

## How Language Models Work - Simple Analogy

Imagine you're reading a book with a friend, but some words are covered up:

> "The weather is very ____ today, so I think I'll stay inside."

You can guess the missing word might be "cold", "hot", "rainy", or "bad" based on the context.

## How Language Models Work - Simple Analogy

Language models do exactly this, but they:

1. Have read millions of books and websites
2. Can consider much longer contexts
3. Can predict not just one word, but generate entire texts by predicting words one after another.

## The Prediction Process

```
Input: "বাংলাদেশ বাংক আজ"

Model thinks: "What word usually comes after these words?"

Output: "ঘোষণা" (announcement)
```

## Small Language Models vs Large Language Models

Parameter Count Comparison:

**Small Language Models (SLM):** few million - few billion  parameters (roughly)

**Large Language Models (LLM):** few billion - 1 trillion+ parameters (roughly)

## What are Parameters?

Parameters are like the **"memory cells"** of the model. Each parameter stores a small piece of learned information about language patterns.

In reality, parameters are the internal values that the model learns during training. These values, often represented as **weights and biases** in the neural network, determine how the model processes input and generates text.

## A Fun Analogy?

If you think of parameters as neurons in a brain:

**More neurons** = more capacity to learn and remember

**But also** = more energy needed and slower thinking

## Why Build Small Models?

Advantages of Small Models:

1. **Speed:** Generate text much faster
2. **Cost:** Require less computational power
3. **Customization:** Easier to train and build on specific domains (like Bangla financial news)
4. **Understanding:** Simpler to study and learn from
5. **Deployment:** Can run on regular computers, not just powerful servers
6. **Efficiency:** Enables real-time applications on edge devices (e.g., phones, IoT, embedded systems)
7. **Privacy:** Critical for privacy-sensitive applications like healthcare, smart home systems, and personal AI assistants.

## Trade-offs:

- Less general knowledge
- May struggle with very complex tasks
- Shorter coherent text generation.
- Outputs may be brittle or overconfident on out-of-distribution data.
- Tend to overfit small datasets or struggle with generalizing to unseen prompts.
- Diminishing returns as you push SLMs with better data or tuning—some tasks are fundamentally bottlenecked by parameter count.

## Our Project Goal

### What We're Building:

- **Model Size:** 58 million parameters (small but capable)
- **Language:** Bangla (Bengali)
- **Domain:** Financial news articles
- **Task:** Generate coherent Bangla financial news text

## Why This Specific Project?

- **Bangla Language:** Underrepresented in AI, important for 300+ million speakers
- **Financial Domain:** Structured language, good for learning
- **Manageable Size:** Can train in a few hours on Google Colab
- **Complete Pipeline:** Learn every step from data to generation

## Expected Outcomes

By the end of this course, your model will be able to:

```
Input: "বাংলাদেশ বাংক আজ"
Output: "বাংলাদেশ বাংক আজ নতুন সুদের হার ঘোষণা করেছে যা
আগামী মাস থেকে কার্যকর হবে।" ( Placeholder )
```

## Environment Setup

### Learning Objectives

You will:

- Set up Google Colab for deep learning
- Understand GPU acceleration basics
- Install and import all required libraries

## Google Colab Setup

### What is Google Colab?

Google Colab is a free cloud-based platform that provides:

- **Jupyter Notebooks:** Interactive coding environment
- **Free GPU Access:** Powerful graphics cards for training
- **Pre-installed Libraries:** Most ML libraries already available
- **Google Drive Integration:** Easy file storage and sharing

## Getting Started with Colab

### Step 1: Access Colab

1. Go to colab.research.google.com
2. Sign in with your Google account
3. Click "New Notebook"

### Step 2: Enable GPU

1. Go to Runtime → Change runtime type
2. Select "GPU" from Hardware accelerator
3. Click "Save"

## Step 3: Test GPU Access

```
import torch
print("CUDA available:", torch.cuda.is_available())
print("GPU name:", torch.cuda.get_device_name(0) if
torch.cuda.is_available() else "No GPU")
```

**Expected Output:**

```
CUDA available: True
GPU name: Tesla T4
```

## Colab Notebook Basics

### Cell Types:

**Code Cells:** For Python code (press Shift+Enter to run)
**Text Cells:** For documentation and explanations

**Important Shortcuts:**

- **Ctrl+M B:** Add cell below
- **Ctrl+M A:** Add cell above
- **Ctrl+M D:** Delete cell
- **Shift+Enter:** Run cell and move to next

**File Management:**

- Files uploaded to Colab are temporary
- Use Google Drive for permanent storage
- Session resets after 12 hours of inactivity

## Understanding Our Dataset

**Learning Objectives:**

**You will:**

- Understand HuggingFace Hub and authentication
- Explore the Bangla Financial News dataset
- Understand why this dataset is suitable for language modeling

## What is HuggingFace?

- **Pre-trained Models:** Thousands of ready-to-use AI models
- **Datasets:** Large collection of datasets for training
- **Tokenizers:** Tools to process text for models
- **Community:** Researchers and developers sharing resources

## Why Use HuggingFace Hub?

- **Free Access:** Most datasets and models are free
- **Quality:** Curated and well-documented resources
- **Easy Integration:** Simple Python APIs
- **Version Control:** Track changes and updates
- **Community Support:** Active community for help

## Authentication Setup

### Step 1: Get HuggingFace Token

- Go to huggingface.co
- Create a free account
- Go to Settings → Access Tokens
- Create a new token with "Read" permissions
- Copy the token

## Step 2: Add Token to Colab

```
from google.colab import userdata

# Add your token to Colab secrets
# Go to the key icon (🔑) in the left sidebar
# Add a new secret named 'HF_TOKEN' with your token value

# Test access
hf_token = userdata.get('HF_TOKEN')
print("Token loaded successfully!" if hf_token else "Token not found!")
```

## Step 3: Login to HuggingFace

```
from huggingface_hub import login

login(token=userdata.get('HF_TOKEN'))
print("Successfully logged in to HuggingFace!")
```

## Bangla Financial News Dataset

### Dataset Overview

- **Repository:** ashtrayAI/Bangla_Financial_news_articles_Dataset
- **Content:** Bangla financial news articles
- **Size:** 7,695+ CSV files
- **Language:** Bengali/Bangla
- **Domain:** Financial news and reports

## Why This Dataset?

### Advantages for Language Modeling:

- **Consistent Style:** News articles have similar structure
- **Rich Vocabulary:** Financial terms expand model knowledge
- **Coherent Text:** Well-written, grammatically correct
- **Domain-Specific:** Useful for financial applications
- **Bangla Language:** Addresses underrepresented language

## Dataset Exploration

### Step 1: Examine Repository Structure

```
from huggingface_hub import list_repo_files

# List all files in the repository
repo_id = "ashtrayAI/Bangla_Financial_news_articles_Dataset"
files = list_repo_files(repo_id=repo_id,
repo_type="dataset")

# Show first 10 files
print("First 10 files:")
for i, file in enumerate(files[:10]):
    print(f"{i+1}. {file}")

print(f"\nTotal files: {len(files)}")
```

## Step 2: Filter CSV Files

```
# Focus on CSV files in the specific folder
folder = "Bangla_fin_news_articles"
csv_files = [f for f in files if f.startswith(folder + "/") and
f.endswith(".csv")]

print(f"Found {len(csv_files)} CSV files in {folder}/ folder")
print("\nFirst 5 CSV files:")
for i, file in enumerate(csv_files[:5]):
    print(f"{i+1}. {file}")
```

## Step 3: Sample Data Examination

```
from huggingface_hub import hf_hub_download

# Download and examine one CSV file
sample_file = csv_files[0]  # First CSV file
local_path = hf_hub_download(
    repo_id=repo_id,
    filename=sample_file,
    repo_type="dataset",
    token=userdata.get('HF_TOKEN')
)

# Load and examine the data
sample_df = pd.read_csv(local_path)
print("Sample DataFrame Info:")
print(f"Shape: {sample_df.shape}")
print(f"Columns: {list(sample_df.columns)}")
print("\nFirst 3 rows:")
print(sample_df.head(3))
```

## Sample Data Characteristics:

```
# Analyze text lengths
if 'News' in sample_df.columns:
    text_lengths = sample_df['News'].str.len()
    print(f"Text length statistics:")
    print(f"Average: {text_lengths.mean():.0f}
characters")
    print(f"Min: {text_lengths.min()} characters")
    print(f"Max: {text_lengths.max()} characters")

    # Show a sample article
    print("\nSample article:")
    print(sample_df['News'].iloc[0][:200] + "...")
```

code

## Practical Exercises

### Exercise 1: Environment Test

Create a new Colab notebook and:

1. Enable GPU access
2. Import all required libraries
3. Test GPU availability
4. Create a simple PyTorch tensor on GPU

## Practical Exercises

### Exercise 2: HuggingFace Exploration

1. Create HuggingFace account and token
2. Explore the dataset repository on HuggingFace
3. Download and examine 3 different CSV files
4. Count total number of articles across files

## Practical Exercises

### Exercise 3: Data Analysis

1. Load 10 CSV files from the dataset
2. Combine them into single DataFrame
3. Analyze text length distribution
4. Find the longest and shortest articles
5. Create a simple word frequency count

## Assessment Questions

### Knowledge Check

1. What is the main difference between small and large language models?
2. Why is the Bangla Financial News dataset suitable for our project?
3. Which dataset would you use? What type of SLM would you like to build? Why?

## Troubleshooting Guide

### 1. GPU Not Available

- **Problem:** `torch.cuda.is_available()` returns False
- **Solution:** Runtime → Change runtime type → GPU → Save

### 2. HuggingFace Authentication Failed

- **Problem:** Token not working
- **Solution:** Check token permissions, ensure it's added to Colab secrets correctly

### 3. CSV Loading Errors

- **Problem:** Encoding issues with Bangla text
- **Solution:** Use `pd.read_csv(file, encoding='utf-8')`

## Next Module Preview

In Module 2: Data Preparation Pipeline, we will:

- Download the complete dataset (7,695+ CSV files)
- Combine all files into a single, clean dataset
- Split data into training and validation sets
- Learn about tokenization and convert text to numbers
- Prepare data for efficient model training